RC 20683 (01/07/96)
Computer Science

IBM Research Report

Updating and Structure in

Non-Monotonic Theories

(PhD Dissertation 1992, Stanford & IBM)

Benjamin N. Grosof

IBM Research Division
T.J. Watson Research Center
P.O. Box 704, Yorktown Heights, NY 10598
(914) 784-7783 direct -7455 fax -7100 main
Internet: grosof@watson.ibm.com (alt.: grosof@cs.stanford.edu)
World Wide Web: http://www.research.ibm.com

Research Division
Almaden · T.J. Watson · Tokyo · Zurich

Abstract

The challenge we address is how to support non-monotonic reasoning tasks over large-scale knowledge bases of rich expressive form. We identify a central problem of updating: which parts of the previous retractable conclusions are safe, i.e., unaffected, when various new axioms are added? This problem is important not only for the task of belief maintenance and revision, but also for specification and for inference. The difficulty arises because non-monotonicity implies potential globality of conflicting interaction. We attack the problem at a logical level, choosing the circumscription formalism as our vehicle of analysis for its mathematical convenience and expressive power.

This dissertation provides a primary groundwork for implementing circumscriptive (and other) non-monotonic reasoning systems that go beyond previous ones in several respects: to perform forward inference and to maintain a body of valid conclusions, as well as to answer queries, for more expressively complex and larger-scale theories. Our major results build sequentially. 1) We extend the circumscription formalism by generalizing the idea of prioritization (relative precedence) so as to enable, for example, the more adequate representation of source reliability and of default inheritance networks. 2) We show that non-monotonic theories are hierarchically decomposable in a manner analogous to programming languages with side effects. 3) We demonstrate several broad cases of safeties of updating, including updating with new default rules. Enabling conditions include syntactic independence and positivity, as well as relative prioritization. 4) We define a generalized "assumption-based" truth maintenance scheme to support inference and belief revision.

For more detail: see section 1.4 (outline) and section 9.1 (conclusions).

About This Document; Follow-On Work

This is a reformatted and slightly revised version of the author's PhD Dissertation, dated October 26, 1992, Stanford University Computer Science Department (see address below). The author completed this dissertation while working at IBM Research full-time during 1988–92.

For follow-on work and papers, see IBM Research's World Wide Web site (URL http://www.research.ibm.com), navigating especially to:

- Benjamin Grosof's personal home page
  (currently URL http://www.research.ibm.com/people/g/grosof/home.html); and/or

- Publications, then CyberJournal which contains on-line Research Reports
  (currently URL http://www.research.ibm.com:8080)

The original version of the dissertation is available through University Microfilms, Inc., of Ann Arbor, Michigan, USA.

In addition, this dissertation may be available as a Technical Report or Technical Note through Stanford:

Computer Science Department, Margaret Jacks Hall (Building 460),
Stanford University, Stanford, California 94305, USA;
World Wide Web URL http://www-cs.stanford.edu ;
Telephone (415) 723-2273; Fax: (415) 725-7411;
E-mail: publications@cs.stanford.edu .

Revisions and Corrections in this Version

Aside from reformatting, the revisions in this version relative to the original version of the dissertation are twofold. The first is different preface sections (the title and 350-word abstract are unchanged, however). The second is the following.

Corrections of minor typos and inadvertent omissions:

- page 69: Added footnote explanation of treatment of the "tie case" in Theorem 2.68.
- sections 7.5.4 and 7.5.5: Corrected the appearances of the denotation operator (') in notation describing propositions, implications, and entailments.
- page 270: inserted "want" before "to add extra data structures to facilitate contents queries."
- page 349: Corrected cross-reference to Corollary 3.46 and inserted "rather" before "than".

Contents

List of Figures

Acknowledgements

Many people have provided comments, encouragement, and helpful discussion: too many to name. I am sure that I have forgotten some below — to them, my apologies, but you and I know who you are, nevertheless.

I would particularly like to thank:

- First and foremost, my principal adviser Nils Nilsson, my co-adviser Michael Genesereth, and my technical adviser Vladimir Lifschitz. They have inspired me to do AI, and have been a fount of helpful suggestions, tough criticism, encouragement, and support. They have taught me by questioning, as well as by example, how to do research.
- IBM, the company, for its wonderful support of my finishing my dissertation while working at Thomas J. Watson Research Center.
- IBM, the people, for the same: especially, Daniel Sabbah, Sanjaya Addanki, and Wlodek Zadrozny.
- The General Electric Foundation, the National Science Foundation, and the Fannie and John Hertz Foundation for generous, full graduate fellowship support.
- My mentors in AI while I was at Stanford: especially, John McCarthy, for introducing me to circumscription and teaching me by example how to think deeply about thinking; Matt Ginsberg, for being so critical; Bruce Buchanan for first interesting me in AI; Ed Feigenbaum, for helping to make AI such a big world at Stanford; and Doug Lenat for being so can-do.
- My mentors in non-AI computer science at Stanford: especially, Bob Floyd and Jeff Ullman.
- Devika Subramanian and Stuart J. Russell: for countless discussions, sharing enthusiasm, encouragement, and being such good friends.
- My colleagues and friends at IBM: especially, Leora Morgenstern, Hector Geffner, Sridhar Mahadevan, and Francisco Corella.
- The members of the Non-Monotonic Reasoning Seminar, MUGS, and PRINCIPIA at Stanford in 1984–1988.

- Ron Loui, Kurt Konolige, Henry Kautz, Bart Selman, Gerd Brewka, John Pollock, Andrew Baker, Peter Rathmann, Nicolas Helft, Michael Wellman, Jon Doyle, David Poole, David Etherington, Ray Reiter, Johan de Kleer, Narinder Singh, Eric Horvitz, David Heckerman, Haym Hirsh, Daniel Lehmann, Sarit Kraus, Ron Brachman, Hector Levesque, and Ron Kohavi for their help and interest.

- My friends, who helped keep me sane: especially, Joan Feigenbaum and Jeff Nussbaum, who have been like family to me; the Lindauers; Olivier Lichtarge and Karen Urbani; and the Targs.

- My professors at Harvard University when I was an undergraduate: Howard Raiffa and Yu-chi Ho, for stoking my interest in the decision sciences; and Eric Teicholz and Harry Lewis, for first provoking my interest in computer science.

- Sara Merryman, for helping to make the adminstrative process so smooth.

- My deepest thanks goes to my family, to whom I dedicate this work with love. To my mother Miriam who has inspired me to be an intellectual and to give. To my wife Janine who has inspired me to stretch to be the most I can, and to keep the rest in perspective. To my brother David who has always kept me on my toes. And to all the rest of my family, who have borne with me throughout the arduous process.

Chapter 1

Introduction

Guide to Reader: Section 1.4 includes an overview of this chapter.

1.1 Ultimate Objectives

The overall aim of our research, and that of many others in artificial intelligence (AI), is to discover how to construct autonomous intelligent agents equipped with general reasoning abilities.

As AI has progressed, it has been recognized that an important goal is to get computers to believe that they can be wrong. Old science fiction movies are full of robots that, when their expectations are violated, wail "Does Not Compute!" and pour smoke from their ear-equivalents. Today's computer users are all too familiar with the blank arrogance that usually accompanies error by the machine. Yet the solution is not over-caution. Those old flicks are also full of computers that spew oracular wisdom until, in an emergency, they torpidly bleat "Insufficient Data.". The heroes are left to rediscover their own powers of decision and, thereby, to reaffirm their uniquely human qualities.

Unlike Hollywood, we would like our robots to emulate the best of human intellection: to be humble, open-minded, yet bold. We want them conscious of their own fallibility, but able to act when information is partial. Like humans, we want them to draw tentative conclusions, and to revise them when they receive new information.

1.2 Non-Monotonic Logical Formalisms

We, along with many others in AI, attempt to view reasoning "declaratively": as inferences in a logical system.[1]

[1]For some discussion of the declarative approach in AI and its advantages, see [Genesereth and Nilsson, 1987] (chapter 1) [Nilsson, 1983] [Doyle, 1985] [McCarthy, 1987]. Briefly: in (what we mean by) the declarative approach, one views the sequence of transformations that a program performs on its data structures as inference steps in a logical system. The logical sentences involved in these steps are the attributed meanings of the corresponding data structures. Declarativism is just one broad methodological approach among several in AI. Some of its advantages are

Logically speaking, human commonsense inference is *non-monotonic*: adding new premises may, in general, cause previous conclusions to be retracted. Humans frequently use rules that are subject to exceptions: *default* information. By contrast, classical logic is monotonic: adding new premises never leads to retracting previous conclusions. To paraphrase "Love Story"[2]: in classical logical systems, proving a theorem means not ever having to say you're sorry.

Over the last fifteen years, a sub-community of AI has developed that studies default and non-monotonic reasoning declaratively. Researchers have attempted, with some success, to capture pervasive patterns of human inference by creating non-monotonic logical systems that formalize and simulate these patterns, albeit in an idealized fashion that ignores many subtleties. Applications of these non-monotonic logical systems have been discovered in other areas of computer science, e.g., the semantics of logic programs and of databases.

In this thesis, we are especially interested in formalisms for default reasoning that are relatively expressively rich and that are equipped with a relatively strong, model-theoretic semantics. In many challenging applications, expressive richness is required to represent the kind of beliefs and reasoning involved (see section 1.6). To reap the full advantages of a declarative approach requires a strong semantics: so that reasoning steps can be described as inferences that are semantically valid with respect to the logical system.

Some examples of the formalisms we have in mind when we say "expressively rich" are: circumscription [McCarthy, 1980] [Lifschitz, 1984], Default Logic [Reiter, 1980], Autoepistemic Logic [Moore, 1985], Non-Monotonic Modal Logic [McDermott and Doyle, 1980], model-preference logic [Shoham, 1988] [Brown and Shoham, 1989] [Boddy *et al.*, 1989] [Selman and Kautz, 1989b], and Ginsberg's Multi-Valued Logic [1988]. By contrast: Prolog-style logic programs [Kowalski, 1979] [Clocksin and Mellish, 1981], "justification-based" Truth (Reason) Maintenance Systems [Doyle, 1979], and path-based inheritance systems [Touretzky, 1986] are more expressively limited and/or are not equipped with as strong a semantics.

Terminology: In the remainder of the thesis, for brevity's sake, we will simply refer to such formalisms as "**expressively rich**". View this as shorthand for "relatively expressively rich and equipped with relatively strong semantics". To qualify as expressively rich, we require, firstly, that the formalism be able to express both for-sure and default beliefs of arbitrary first-order logical form.

---

that it:

- cleanly separates beliefs and premises from inferential processing;
- helps unify and integrate both procedures and data across systems;
- aids concise communication and understanding;
- provides semantics — therefore, a check on intuition, and formal guarantees enabling partial validation;
- facilitates incremental modifiability of programs, due to the existence of standard implemented inference systems and re-usable canonically-interpreted knowledge bases. One may update a program with new axioms, or with new inference mechanisms.

[2]the novel by Erich Segal (NY: Harper & Row, 1970), containing the (?in)famous line "Love means not ever ..."

Secondly, we require that it have a semantics which constrains the interpretations of these beliefs in a significant manner (e.g., beyond very simple direct contradictions) roughly comparable to classical logic. Justification-based TMS's cf. [Doyle, 1979] and inheritance networks cf. [Touretzky, 1986], by contrast, treat each belief essentially as an opaque node. There, belief in p and in $p \supset q$ does not, in general, constrain q to be believed. That is weaker than what we are looking for. Naturally, there is a spectrum of expressive richness, especially when one considers special-case classes of the expressively rich formalisms.

(In the non-monotonic reasoning literature, there are now many known close relationships and equivalences among the above-mentioned expressively rich formalisms; this provides evidence that the "expressively rich" category is a fairly coherent notion.)

Unfortunately, there is still a long way to go in applying non-monotonic logical systems to help achieve generally reasoning agents. The current state of the field is that the expressively rich formalisms, in particular, have not yet found much implemented application. Almost always in the literature, the scale of the example theories that have received attention has been so small (roughly, ten default axioms or less) as to be "toy" from an ultimate application perspective. This has led to some criticism and soul-searching in the non-monotonic reasoning community.[3]

1.3 Problems Addressed, Methodology, and Nature of Results

Our goal in this work is to make progress on the usual two dimensions of knowledge representation:

1. **Expressiveness:** To say what one wants to say. The logical system should enable one to express beliefs of adequately complex form. It should facilitate expressing the premises and the conclusions in a manner that corresponds relatively closely to, respectively, what one "knows" and to what one thinks in terms of. The logical system should enable one, by asserting premises, to concisely and implicitly specify exactly the desired set of sanctioned inferences (no more and no less).

2. **Mechanization:** To actually accomplish the reasoning, automatically. Inference and updating should be achievable with procedures that are relatively efficient computationally.

We address both of these dimensions, in tandem, for non-monotonic reasoning, declaratively using the tool of expressively rich non-monotonic logical formalisms.

More particularly, we are most interested in a class of (potential) applications that we will call **learning agents (terminology)**. By this, we mean agents that, dynamically and on a continual basis, acquire new premise beliefs and revise a body of (non-monotonically) concluded beliefs accordingly.

[3] e.g., panel discussions at the First International Conference on Principles of Knowledge Representation and Reasoning (held Toronto, Ontario, May 1989), the Third International Workshop on Non-Monotonic Reasoning (held in South Lake Tahoe, California, June 1990), and the AAAI Spring Symposium on Logical Formalizations of Commmonsense Reasoning (held Stanford, California, March 1991)

We identify and address two major problems. The first is the **need to generalize the expression of prioritization-type precedence between defaults**. Precedence is a kind of relative confidence.[4]

The second major problem is the **need for locality in performing inference and updating**. We attack this problem of **scale**, in domain-independent fashion, **at the logical level**. We develop a **divide-and-conquer strategy** that relates **decomposable structure** in sets of premise beliefs to decomposable structure in sets of conclusions; hence, the title of the thesis. This strategy is a new one for the field of non-monotonic reasoning.

Methodologically, we pick one particular non-monotonic formalism to work with: circumscription [McCarthy, 1980] [Lifschitz, 1984] [McCarthy, 1986]. Circumscription is one of the historically central formalisms, and has some other advantages. Mathematically, it has been relatively well-studied, and it is close to standard classical logical systems.

This thesis is a **theoretical study**. Our major results take the following form.

- We improve (generalize) the expressive class, with respect to relative precedence ("prioritization") of default premise beliefs. This is motivated by the kind of information available in applications. It is important and useful especially when there are lots of defaults that "conflict" (compete) with each other. We apply this generalization to circumscription, and to some other formalisms as well. We show that our generalization of prioritization is in several respects the most general in the non-monotonic reasoning literature.

- We aid the modular specification of default beliefs, in two ways. Firstly, we develop the idea of composition of prioritization; this is also useful for studying locality. Secondly, we develop a meta-language (the "Circumscriptive Language of Defaults") for specifying circumscriptions which enables defaults and prioritization to be specified directly and incrementally. This makes specification more natural, i.e., expressively convenient. It also aids the study of updating.

- We develop ideas of decomposition, hierarchy, scope, and external-context declarations in the non-monotonic logical system, in a manner analogous to programming languages with side effects. We use prioritization, which mediates conflict, as one key to the structure of theories.

- We find local (well-behaved, modular) cases of reasoning, with the opportunity to be exploited algorithmically in tractable fashion. These cases offer improved efficiency relative to more general-class reasoning, and thus offer a tempting target for expressive restrictions in specification.

- We develop a new "genealogical" style of architecture for non-monotonic reasoning, which exploits our results about locality in inference and updating. This architecture revolves

[4]See sections 1.5 and 2.4 for review of the concept of prioritization-type precedence.

around a new kind of "assumption-based"[5] truth maintenance system. The genealogical architecture is especially suitable for learning agents.

- We design, and/or aid the design, of procedures, where previously there have been none, for several kinds of expressively rich inference tasks:
  - forward (bottom-up) reasoning, not just backward (top-down) reasoning [6];
  - belief maintenance and updating, including updating with new defaults;
  - reasoning over a new, generalized (prioritization) expressive class.

  As part of this, we identify opportunities for concurrency and alternative sequencings.

We conceptualize and investigate theoretical properties of structure, and of updating, inference, and specification, primarily in non-monotonic theories expressed in circumscription. We believe that the problems we attack, and much of the behavior of the circumscriptive formalism, are common to most logical systems for default reasoning. We show how some of our results and ideas map over to other expressively rich non-monotonic formalisms. We conjecture that many more of our results and ideas will map over; more and more is becoming known about commonality relationships of circumscription to many other such formalisms. We give high-level procedures and results that appear useful for designing procedures, but leave for future work the development of detailed algorithms. We give some computational complexity analyses, but leave for future work the development of many other complexity implications of our results.

By conceptualizing and analyzing parts of non-monotonic (circumscriptive) theories, we show how to preserve soundness while drawing only part of the overall set of valid conclusions, for a variety of new cases. (Previous results were just about query-answering.) This provides, indirectly, some principled basis for compromise with resource limitations. Circumscription, and all of the other expressively rich approaches we listed in section 1.2, are tools to describe reasoning of an idealized kind: freed from resource constraints. They also do not provide a way to describe differential value of inference. To address resource limitation more directly is an important direction for future work. More generally, how to *control* default inference is a vital question for future work.

We view the results in this thesis as initial steps on a road. The history of first-order logic provides a perspective. Quite a lot of work was required to understand logical properties, invent alternative proof procedures, and find exploitable expressive classes, before the relatively powerful inference methods now available, e.g., underlying today's Prolog-style logic programming systems, could arise.

[5] in the sense of de Kleer [1986a]

[6] By "backward" reasoning, we mean reasoning that works backward from a goal query, attempting to prove it by forming sub-goals. By "forward" reasoning, we mean reasoning that works forward from premises and/or previous inferences to create new inferences in a bottom-up and less goal-directed manner.

1.4 Outline of Thesis

In section 1.1, we described our ultimate objective: to construct autonomous intelligent agents equipped with general reasoning abilities, including the ability to draw tentative conclusions and to revise them when they receive new information. In section 1.2, we reviewed the tool of expressively rich non-monotonic (NM) logical formalisms. These support a declarative approach towards the ultimate objective. In section 1.3, we laid out our goals more specifically in terms of these formalisms: to improve expressiveness, especially with respect to prioritization-type precedence, and to improve the practical scalability of mechanized, expressively rich, non-monotonic reasoning. We are especially interested in learning agents as a class of potential applications. We outlined our strategy for scale: divide and conquer.

In section 1.5, we review concepts in non-monotonic reasoning (NMR), including defaults and precedence; and we introduce notation, terminology, and examples that we will use throughout the thesis. In section 1.6, we review the applications of non-monotonic reasoning, and elaborate on our notion of learning agents. We argue, with a detailed example, for the importance of expressive generality, especially with respect to the first-order forms of default and for-sure premise beliefs, and with respect to the partial orders of precedence among defaults. We argue, in addition, for the importance of updating with new defaults and new, explicit precedence, not just new for-sure premise beliefs.

In chapter 2, we generalize the formal idea of prioritization in circumscription, in two ways. The first is to permit the precedence partial order to be any well-founded, e.g. any finite, partial order. Previously, precedence was restricted to be "layered" ("stratified") in a fashion similar to military rank; yet our intended applications demand it be non-layered. The second is to enable precedence and prioritization to be defined over groups of defaults, in hierarchical fashion. We go further and show how prioritization can be generalized to apply even beyond circumscription, to the aggregation of arbitrary preference criteria, whether in the realm of NM reasoning or not, as long as the criteria are transitive and reflexive. We show that our idea of prioritization subsumes several previous ideas of precedence in the NMR literature. Along the way, we review and discuss circumscription in rather gory technical detail: in preparation for its use in the rest of the thesis.

In chapter 3, we discuss a variety of issues that arise in defining NM theories in circumscription. Our main focus is to define a new NM formalism, the Circumscriptive Logic of Defaults (CLD). This is essentially a meta-linguistic convention for specifying circumscriptive theories. This NM formalism meets three of our desires that previous variants of circumscription did not. Firstly, defaults and priorities can be specified directly as axioms. Secondly, the policy[7] parameter in circumscription is effectively standardized, making the formalism truly a single NM logical system. Thirdly, the prioritization class is more general, building on the results of chapter 2.

[7]see Definition 2.8

In addition, we give new results about “abnormality” theories, satisfiability, and “fixing”. (Intuitively, fixing is a kind of partial-paralysis restriction on which formulas can be concluded non-monotonically; we discuss it in sections 2.3 and 3.5.) In particular, we show that CLD with fixing is expressively reducible to the predicate case (of prioritized circumscription). We show that well-behavior, including satisfiability, is guaranteed for CLD theories and predicate circumscriptions with non-layered prioritization, whenever they are either universal or “quasi-”propositional (obey weak domain closure). These results underpin the applicability of our new formalism, and are used in our analyses in later chapters.

In chapter 4, we identify problems of scale in expressively rich NMR: in inference, in specification, and in updating, especially belief revision. The remainder of the thesis (especially, chapters 5 through 7) primarily addresses these problems of scale. In chapter 4, we develop an analysis and strategy of attack on scale, in the spirit of divide-and-conquer. We develop develop concepts and motivating observations, especially about decomposition and updating, that we will use in later chapters. In particular, we observe that logical non-monotonicity is a kind of **logical globality (terminology)** or non-modularity. We reformulate the problems of scale in terms of the need for a kind of **locality (terminology)** in inference and belief revision.

Terminology: By a previous conclusion being **safe** after an update, we mean that it does not need to be retracted. By an update being **globally monotonic**, we mean that it requires no retractions for the entire NM theory.

We show that the analogues of many basic properties that one takes for granted in classical monotonic logic, fail to hold in prioritized default circumscription and in many other formalisms for NM / default reasoning. This indicates that characterizing monotonicity of updating even at the level of the logic, computational challenges aside, is a tricky enterprise.

In classical logic, we take for granted a convenient idea of a part of theory. However, logical globality means we have to “work for it” in non-monotonic formalisms. Drawing on first-order logic for inspirtation, we develop two different such concepts of **decomposition (terminology)** into parts: **conjunctive**, and **serial**. We observe that the potential gains from decomposition include locality, selectiveness, and concurrency in inference. We also observe that decomposition can potentially expose safeties of updating, by comparing a decomposition after an update to a decomposition before the update. Our concepts of decomposition are hierarchical: one can decompose into parts, then decompose those parts more finely.

In chapter 5, we develop some general decomposition theorems using priority. Our investigations illuminate the effects, and mathematical complexity, of non-layered prioritization. We introduce the concept of “guarding” to represent the structure of potential conflicting interaction. Guards play a role analogous to that of declarations in programming languages with side effects. We then apply the decompositions to obtain theorems about safeties of updating based on priority.

Finally, we develop some results about the strong locality available for special cases where prioritization is totally ordered. These results include: decompositions; safeties of updating; and a new forward-inference algorithm.

In chapter 6, we develop more results about strong localities available in special cases. In section 5.13, we studied strong localities for special cases defined primarily in terms of prioritization. Here we investigate several special cases defined primarily in terms other than prioritization; our results are interesting even for parallel (i.e., empty) priority. For these special cases, as with total prioritization, we show that defaults are guaranteed not to conflict at all: we develop clean conjunctive decompositions, and strong safeties of updating; some are at coarse grain size and some are at fine grain size. And as with total prioritization, we are able to show that one case collapses to first-order with opportunities for new inference algorithms.

We observe that the bases for locality structure available in these cases fall into two categories: what we call "orthogonality" and "sympathy". The concept of logical orthogonality corresponds to the intuition of passing by on different geometric planes, or of "ships passing in the night". More formally, orthogonality is a kind of logical independence. The concept of sympathy corresponds to the intuition of pulling in the same direction, or working in harmony. More formally, we define directionality in terms of syntactic positivity and negativity. By contrast, with total prioritization, the concept of priority dominance corresponds to the intuition of winning in a battle.

Our special-case locality results in chapter 6, as in section 5.13, build, in two ways, on our general-case canonical decomposition results and methods in chapter 5. Firstly, we use the general-case decompositions **to help prove** the special-case results about decomposition. And we use our general method of **differential decomposition** (from section 4.5) together with those special-case decompositions, to prove special-case safeties of updating. Secondly, general-case decomposition may result in part theories that fall into one or more different special-case classes. Thus the general-case decompositions tell us how to **glue together** special-case results, using them for parts, rather than all, of a global axiom set / theory.

In chapter 7, we discuss how to exploit our results to design procedures for NM specification, inference and belief revision: primarily, in prioritized predicate and default circumscription; and, more generally but to a lesser extent, in other NM formalisms as well. We sketch a new "Genealogical" architectural method for sound (but perhaps incomplete) NMR, e.g., learning agents in the sense discussed in sections 1.3 and 1.6. This architecture is able to exploit results on decompositions and safeties, e.g., ours in chapters 2 through 6. The architecture employs a novel kind of truth maintenance mechanism: we develop this in detail. This Genealogical Truth Maintenance System is an unusual variant on de Kleer's [1986a] Assumption-based TMS: each assumption corresponds to an entire global NM theory or an entire decomposed NM sub-theory. We call the architecture, and its use of truth maintenance, "genealogical" because they revolve around the use of sub-theory relationships: conclusions "descend" to the current global theory from part theories and from previous theories. We show that the Genealogical Truth Maintenance task is computationally tractable: polynomial in the justifications (justifications in the sense of de Kleer), unlike the ordinary ATMS task, which is NP-hard.

We also sketch how to implement most of our major results on decompositions, safeties, specification, reformulations, and inference procedures from chapters 2 through 6. We observe that the computational cost involved, both to recognize the relevant conditions / special cases and to perform the decompositions etc., is polynomial in the size of the global CLD axiom set. Thus, we observe that our Genealogical NMR architecture is practically implementable.

In chapter 8, we discuss several, somewhat miscellaneous, topics. Most of these topics touch upon several of the previous chapters; and most involve discussion of related work.

We sketch how our earlier results imply methods for approximation of NM theories and inference, and contrast these with other approaches to approximation. We discuss how our earlier results ease the task of knowledge acquisition. We compare our approach to axiomatizing policy in circumscription to that of Lifschitz [1988a] and show a number of advantages over his variant of circumscription. We discuss related work by Rathmann [1990] about combining NM theories in his variant of circumscription. We briefly review several other NM formalisms: Default Logic, Poole's formalism, Autoepistemic Logic, and logic programs. We observe that many of our major results in this thesis apply to each of these formalisms.

In chapter 9, we summarize, discuss our conclusions, and suggest directions for future work.

In the Appendix, we give the longer and more technical proofs.

Guide to Reader: It is important to **READ CHAPTER A of the Appendix BEFORE READING ANY OF THE PROOFS IN THE APPENDIX!!!** Chapter A gives crucial overview and notation-type information.

1.4.1 Dependencies of Chapters 5 through 7 on Chapters 2 and 3

This sub-section is intended as **Guide to the Reader**.

Chapters 2 and 3 mainly focus on expressiveness issues rather than problems of scale and methods for locality. However, several ideas and results from chapters 2 and 3 directly help with, or impinge upon, our investigation of scale and locality and updating. Some of the most important such relationships are:

- In chapters 2 and 3, we give expressive generalizations of circumscription, mainly with respect to priorities, that define the expressive classes investigated in chapters 5 through 7.

- In chapter 3, we define a formalism for the updating of the policy parameter (priorities, defaults / minimized predicates, "fixtures") in circumscription; this defines the terms in which we study scale, especially updating, in chapters 5 through 7.

- In chapter 3, we study "fixing", which is intuitively a kind of partial-paralysis restriction on what non-mononotonic conclusions are permitted. Fixing arises in the course of our later decomposition results in chapters 5 through 7, and is, generally, an important feature of specification in circumscription.

- In chapter 2, we develop the concept of composing prioritization. This turns out to be a useful tool for hierarchical decomposition and, generally, for analysis, in chapters 5 through 7.

- In chapter 2, we give a global monotonicity result for prioritization updates. This is also used often in the proofs of results in chapters 5 and 6.

- In chapter 3, we give a global monotonicity result for default and for-sure updates that are fixed in a particular sense.

In addition, of course, our definitions and notation about circumscription, from chapter 2, are used throughout the rest of the thesis.

1.5 Non-Monotonicity: Concepts

In this section, we give a somewhat generalized review of the basic concepts of non-monotonic reasoning, and introduce terminology and notation used throughout this thesis. Included are the ideas of logical non-monotonicity, default belief, conflict among defaults, and precedence-type preference among defaults.

For additional background on non-monotonic reasoning, three good starting points are [Reiter, 1987a], [Ginsberg, 1987] (especially the Introduction), and [Genesereth and Nilsson, 1987] (chapter 6).

Relative to a particular logical system, and a set of premises, i.e., *axioms*, we define the associated *theory* as the set of all conclusions that are consequences of those axioms, in the system. We can view the set of conclusions as a function of the set of axioms.

Classical logics (e.g., propositional logic, first-order logic, second-order logic) and the standard modal logics (e.g., S5), are logically monotonic. Logical monotonicity means that as the set of axioms gets larger (in the sense of non-strict set inclusion), then the set of conclusions also does. (Alternative view: in classical logics, as the axiom set grows, the set of its models gets smaller.)

Definition 1.1 (Logical Monotonicity)

For a particular logical system $\mathcal{S}$, let F be the function that takes a set of axioms in $\mathcal{S}$ as input, and produces as output the set of conclusions sanctioned by $\mathcal{S}$. We say that the logical system $\mathcal{S}$ is *monotonic* when

$$\forall \mathcal{A}_1, \mathcal{A}_2. \quad \mathcal{A}_1 \subseteq \mathcal{A}_2 \;\; \Rightarrow \;\; F(\mathcal{A}_1) \subseteq F(\mathcal{A}_2)$$

where $\mathcal{A}_1$ and $\mathcal{A}_2$ are axiom sets. In other words: *Logical monotonicity* $\stackrel{\text{def}}{\equiv}$ *as axioms accumulate, so do conclusions.*

Logical non-monotonicity is, simply, the lack of logical monotonicity.

Definition 1.2 (Logical Non-Monotonicity)

Logical non-monotonicity $\stackrel{\text{def}}{\equiv}$ *as axioms accumulate, conclusions may be retracted.*

Terminology and Abbreviations:
"Non-monotonic logical system" is synonymous with "non-monotonic logical formalism".
"NM" stands for "non-monotonic";
"NMR" stands for "non-monotonic reasoning". Unless explicitly made clear to the contrary, when we say "non-monotonic reasoning", we mean reasoning in a non-monotonic logical system.
Notation: Axiom Sets and Theories:
We will write $\mathcal{A}$ to denote a set of axioms, i.e., an *axiom set*, in some non-monotonic logical formalism. We will write $\mathcal{NMTH}(\mathcal{A})$ to denote the non-monotonic theory, i.e., the set of conclusions, specified by the axiom set $\mathcal{A}$ in that non-monotonic logical formalism.

Our main focus in this thesis will be on reasoning with *defaults*. In formalizing default reasoning, we view premise-type beliefs (axioms) as falling into two status categories: for-sure, or "hard"; and default, or "soft". A for-sure axiom, e.g.

$\bullet\!> \; bird(Tweety)$

is just the same as in ordinary classical logic: it asserts that something (e.g., $Tweety$ is a $bird$) is always true. It is absolute, not tentative. A default axiom, by contrast, should be viewed as asserting that something is true presumptively (normally, typically), but is susceptible to exception. A default axiom specifies a *defeasible* (defeat-able) premise belief; a for-sure axiom specifies a *non-defeasible* premise belief. Here, we will intuitively interpret a default axiom, e.g.,

$:> \; flies(Tweety)$

as asserting that something (e.g., $Tweety\ flies$) is true if the something is consistent. This consistency, however, is relative to the context of the rest of the axioms that are present in the current axiom set.

Different non-monotonic logical systems formally define the idea of a default in different ways. In chapters 2 and 3[8], we will discuss how to formalize the idea of a default in circumscription. For now, however, we will leave it a bit vague, so as not to obscure the essential intuitions by plunging into technical details. Many other formalisms for default reasoning display similar behavior, as we will also discuss later in sections 8.5 and 8.6.
Notation and Circumscriptive Formalism:
In the remainder of chapter 1, we will be discussing examples and concepts in default reasoning. You should view the notation as standing for the Circumscriptive Logic of Defaults, a meta-language that we have invented for specifying default theories in circumscription. We will make the meaning of this notation precise later, in section 3.2, after introducing some generalization of circumscription. For now, we proceed at an intuitive, hand-wave-y level.

Next, we review a classic pedagogical example of default reasoning: Quakers and Republicans. Here, we do it informally; in section 2.4, we formulate it more precisely in circumscription.

[8]See especially Definition 2.11 (default pre-order) at the end of section 2.2, and section 3.2 (the Circumscriptive Language of Defaults).

Example 1.3 (Non-Monotonicity — Quakers and Republicans)
Suppose one's premise beliefs consist of two assertions: firstly, that $Nixon$ is a $Republican$ with for-sure belief status; and, secondly, that, by default, $Republican$s are not $Pacifist$s. More formally, let us consider an example of a starting set of axioms (containing exactly):

$$\mathcal{A}_0 :$$

$$\bullet\!\!> \; Republican(Nixon)$$
$$(d1) :> \; Republican(x) \supset \neg Pacifist(x)$$

Notation:
We introduce here the notation for axioms that we will follow throughout the thesis. The prefix symbol $\bullet\!\!>$ indicates that the following sentence part is to be construed as an assertion with for-sure status. The prefix symbol $:>$ indicates that the following formula part is to be construed as an assertion with default status. A default axiom's formula part may take the form of an open formula (as above), or a closed formula (sentence). When open, it can intuitively be viewed as a schema: the collection of its instantiations.[9] We will see the role of the default axiom's parenthesized label ($d1$ above) shortly.

In our circumscriptive formalization of default reasoning (and, analogously, in many other default formalisms), from this axiom set, one can validly draw the conclusion that $Nixon$ is not a $Pacifist$. Intuitively, since there is no other information beyond these two axioms, it is consistent to presume that the default successfully applies to $Nixon$. More formally:

$$\mathcal{A}_0 \;\;|\!\approx\;\; \neg Pacifist(Nixon)$$

More notation: $|\!\approx$ stands for non-monotonic entailment, i.e., the consequence relation in our non-monotonic logical system. This notion of entailment is "skeptical" (rather than "credulous"). **Terminology:** Briefly: **credulous** entailment means there are, in general, several possible different alternate, mutually exclusive, belief sets that each are sanctioned from a given set of premises; these can be viewed, intuitively, as alternate scenarios. By contrast, **skeptical** entailment means there is always only one belief set sanctioned from a given set of premises. See page 23 for more discussion of skepticality vs credulity.

Suppose that, next, one receives the update that $Nixon$ is for-sure a $Pacifist$, i.e., the new axiom:

$$\mathcal{U}_1 : \qquad \bullet\!\!> \; Pacifist(Nixon)$$

Then the previous conclusion is **retracted** in the resulting, new non-monotonic theory:

$$\mathcal{A}_0 \& \mathcal{U}_1 \;\;|\!\not\approx\;\; \neg Pacifist(Nixon)$$

[9]somewhat similarly to a normal default rule, without prerequisite, in Default Logic [Reiter, 1980]; see [Lifschitz, 1990b] for a precise formalization of this view

(**Notation**: & stands for union, among axiom sets.)
That previous conclusion was **non-monotonic**, i.e., (potentially) **retractable**, i.e., **defeasible (terminology)**. Intuitively, the hard information that $Nixon$ is indeed a $Pacifist$ overrides the soft argument that he is not.

Terminology: We distinguish between conclusions that are potentially retractable, and those that are **non-retractable**, or **monotonic**, in the sense that no update will cause them to be retracted. Any conclusion which can be inferred purely from for-sure axioms is non-retractable.

Example 1.4 (Conflict)
(Continuing the previous example.) Suppose that instead of $\mathcal{U}_1$, one receives the update:

$$\mathcal{U}_2 :$$

$$\bullet\!> \ Quaker(Nixon)$$

$$(d2) :> \ Quaker(x) \supset Pacifist(x)$$

Then

$$\mathcal{A}_0 \& \mathcal{U}_2 \ \not\approx \ \neg Pacifist(Nixon)$$

but also

$$\mathcal{A}_0 \& \mathcal{U}_2 \ \not\approx \ Pacifist(Nixon)$$

The previous conclusion is retracted *without* being able to conclude its negation, as one did above. This is an example of a *pure retraction* (pure defeat).

Intuition: Essentially what is going on here is that the two defaults **conflict**. The $Republican$ default tells one to conclude $\neg Pacifist(Nixon)$, while the $Quaker$ default tells one to conclude the opposite. Since one has no way to resolve the conflict, one can conclude neither. This illustrates a broader phenomena: two or more defaults may conflict, given the for-sure information, in the sense that one cannot consistently believe them all as conclusions. We will discuss the concept of conflict more in section 2.4. [10]

A possible basis for adjudicating conflicts is the idea of *preference* among defaults, in the sense of **precedence**. In many applications of default reasoning, such precedence information is available, and thus it is desirable to represent it in the formalism: several formalisms do. Intuitively, one possible basis for such precedence is that one default originates from a more reliable, e.g., fresher or more authoritative, source of information than another. A second basis is that one default covers a more specific case than another (e.g., in inheritance hierarchies), and represents an overriding exception.

Example 1.5 (Precedence)

[10] Historical Note: Reiter and Criscuolo [1981] is an early study of this issue of conflicting defaults in expressively rich NMR. They investigated it in Default Logic.

(Continuing the previous example.) However, suppose one adds an axiom specifying a preference for one default over the other:

$$\mathcal{U}_3 : \qquad \mathcal{PREFER}(d1, d2)$$

Intuitively, this axiom says that one prefers the *Republican* default over the *Quaker* default, so that in case of conflict between their conclusions, the *Republican* default "wins" by having its conclusion dominate. Intuitively, this might be based on having greater confidence in the *Republican* default than in the *Quaker* default. Then

$$\mathcal{A}_0 \& \mathcal{U}_2 \& \mathcal{U}_3 \quad \models\!\!\approx \quad \neg Pacifist(Nixon)$$

(Compared to a default axiom, the above $\mathcal{PREFER}$ precedence axiom may seem to have perhaps more meta-level flavor, and to look further away from first-order logic. But there is no need to worry. In chapter 2 and section 3.2, we will make all of this precise in terms of circumscription, within the framework of classical logic. See Example 2.29 in section 2.4, and Example 3.9 in sub-section 3.3.1, for circumscriptive formulations of the above example.)

In chapter 2, we will discuss much more the idea of precedence-type preference among defaults. Circumscription can express it with the technical idea of "prioritization". We will discuss intuitive bases for prioritization (section 2.5), and how to generalize it in some ways (section 2.7). We will also discuss how other non-monotonic formalisms express precedence-type preference (section 2.11).

1.6 Motivating Applications of Non-Monotonic Reasoning

In this section, we outline the applications of non-monotonic logical formalisms, and discuss the concerns of expressive generality that especially motivate our work.

1.6.1 The Ubiquity of Non-Monotonicity

Researchers, to date, have identified a host of applications, and potential applications, of non-monotonic reasoning. From one perspective, this is not too surprising: when one takes a declarative approach to describing reasoning, monotonicity is the special case, and non-monotonicity is the more general case.

Here, we briefly outline these kinds of applications. Roughly, we can divide them into two broad "realms".

Representational Conventions:
In the realm of what we call "representational conventions", the non-monotonic logical system is used to help specify a body of conclusions that, in some sense, is a complete collection of knowledge. The non-monotonic logical system makes it possible to "save some breath": it facilitates expressing what one intends to say, in a manner that is concise, and easily modifiable in a natural way. This conciseness is not always just a matter of convenience: often, as with the Closed World Assumption [Reiter, 1978], the non-monotonic logical system makes it possible to give a finite axiomatization of

a theory that would require an infinite axiomatization in a monotonic logical system. For example, suppose one wishes to specify exactly which airline flights are available on a particular airline. Each flight is described as a pair of cities and the departure time. One can easily list the flights in a monotonic logic: using one axiom per flight. More difficult is listing the non-flights. The number of them may be huge, or even infinite: e.g., if time is represented using the real numbers, and/or if there is an infinite domain of possible cities.

Application areas in this realm include:

- the Closed World Assumption, the Generalized Closed World Assumption;
- Logic Programming with negation as failure;
- the default inheritance mechanism in many AI-style Frame-Based Systems [11] and in much of Object-Oriented Programming;

More arguably, also included in this realm are some treatments of:

- Frame Assumptions, e.g., in temporal action and modelling;
- recency precedence in Data Bases (Knowledge Bases).

See [Reiter, 1987a] and [Ginsberg, 1987] (especially the Introduction) for overview and representative articles on these applications.

Empiricism and Action:
In the second realm, what we call "empiricism and action", the non-monotonic logical system is used to specify what an agent should believe when acting, or deciding, in the presence of

- incomplete knowledge; and/or
- changing environment and knowledge.

In these areas of application, the agent's need to act or decide imposes a need to commit to a belief that justifies the action or decision.

Non-monotonic logical formalisms are especially useful as tools for specifying the **dynamics** of belief. In reasoning to support practical action, e.g., in commonsense tasks, an agent's beliefs must evolve dynamically in a fashion that, often, violates monotonicity.

Past and potential application areas in the realm of empiricism and action include:

- inductive theory formation
  - rule and concept learning [Russell and Grosof, 1987] [Grosof and Russell, 1990] [Russell and Grosof, 1990a] [Helft, 1989]

[11] A relatively generic example of such a default inheritance mechanism is described in chapter 13 of [Charniak *et al.*, 1987]. An early progenitor of such systems is: [Bobrow and Winograd, 1977]. Bobrow and Winograd were responding, in part, to the "frames" proposal of [Minksy, 1975].

- perception and sensory interpretation
  - vision
  - motion
  - speech understanding
- natural language understanding and communication [Perrault, 1987] [Appelt and Konolige, 1988] [McRoy, 1989] [Morgenstern and Guerriro, 1991] [Zadrozny, 1988]
- diagnosis [Ginsberg, 1986] [Reiter, 1987b]

- heuristics and flexible policies
- retractable decisions: planning, design
- "Bayesian" and "evidential" reasoning [Grosof, 1988] [Bacchus, 1990] [Bacchus, 1991]
- user modelling
  - intelligent tutoring
  - intelligent user interfaces
- law

Default reasoning is, in these areas, an especially important mode of non-monotonic reasoning. Default beliefs specify a kind of best-guessing policy for the agent to follow. Default-status belief represents the tentative quality of interpretations of empirical information (e.g., in inductive theory formation tasks). It also represents the interactive, context-dependent quality of decision-oriented policy (e.g., in planning, design, and law). Little of our empirically-based knowledge is absolutely for-sure. Little of our decision-making is for-sure. Defeasibility captures this "existential" predicament.

In this thesis, we are especially interested in the realm of empiricism and action, and in default reasoning. However, remarkably, there is a great deal of mathematical commonality between the two realms. The same kinds of non-monotonic logical formalisms are often useful in both realms. For example, an application that originally motivated John McCarthy to develop circumscription was in the realm of representational convention: he was trying to describe the Missionaries and Cannibals problem [McCarthy, 1980]. Vladimir Lifschitz also applied circumscription in this realm: to the semantics of stratified logic programs with negation as failure [1987a]. But we, in other work, have applied circumscription to two different areas in the realm of empiricism and action: inductive rule and concept learning [Grosof and Russell, 1990] (see also [Russell and Grosof, 1987] [Russell and Grosof, 1990a]); and Bayesian reasoning [Grosof, 1988].

Even more particularly, in this thesis, we are most interested in the class of (potential) applications that we call **learning agents**. (Recall terminology on page 3: by this, we mean agents that, dynamically and on a continual basis, acquire new premise beliefs and revise a body of (defeasibly) concluded beliefs accordingly.) In part, this grows out of our other work (mentioned above) on formulating inductive concept learning as non-monotonic inference.

1.6.2 An Example Learning Agent: Dynamic Belief for Practical Reasoning

We will not try here to give you a detailed analysis of how and why default reasoning is useful for all the application areas we listed above in the realm of empiricism and action. Rather, we will try to give you a feeling for its suitability for that realm, via one extended example of a learning agent engaged in a commonsense practical reasoning task.

Example 1.6 (Dynamic Evolution — Commuting)
Let us consider the problem, in rather simplified form, of how Jon plans his daily commute to work. Upon waking in the morning, Jon presumes that his car is OK, and he presumes that the traffic on his usual route to work is also OK. He further presumes that if his car is OK and the (traffic on his usual) route is OK, then he will arrive at work within an hour by driving his car on his usual route. If, today, Jon has no further information upon waking, then he will conclude, by default, that his usual plan will succeed today. All this could be easily represented with initialized values and an if-then statement in almost any medium- or high-level programming language. What's more challenging to represent are the scenarios in which more happens. To wit: Jon also believes that, for-sure, if his car has an oil leak, then his car is not OK. The previous conclusion about his car being OK is overridable: default in status. If, today, Jon learns (exactly) that his car has an oil leak, then the justification for belief in his usual plan's success is removed, and he will retract it. One needs to track the dependency of the conclusions on the defaults used, if one is to reason in an incremental fashion. Therefore, a logical system for default reasoning is useful for this kind of problem.

Next, let's make the problem a bit less simplified. Consider the following elaboration of the story, as a pattern of reasoning about Jon's commute.

> After Jon woke up today, while getting dressed, he believed that his car and route were OK, and, therefore, that, by commuting to work using his usual plan, he would get there within an hour. However, when he went into the kitchen, he saw a note from his wife. It said that his car had an oil leak. Darn! Now, Jon believed that his car was not OK, and, therefore, did not believe his usual plan would succeed. Jon called his wife (who was already at work) and she said that his car did not have an oil leak (she had looked at the car in the garage on her way out this morning; there was no leak visible; leaks are always visible; the note must have been from several months ago and been dislodged when she was cleaning up the counter this morning). Now, Jon believed that his car was OK after all, and that his usual plan would succeed. Breathing a sigh of relief, he settled down to eat breakfast and turned the radio on to his favorite news station. He heard a traffic report that said there was a nasty multi-car accident blocking his usual route. Rats! Now, Jon believed that the route was not OK, and, therefore, no longer believed that his usual plan would succeed. (He would have to take a different, and longer, route, he thought.)

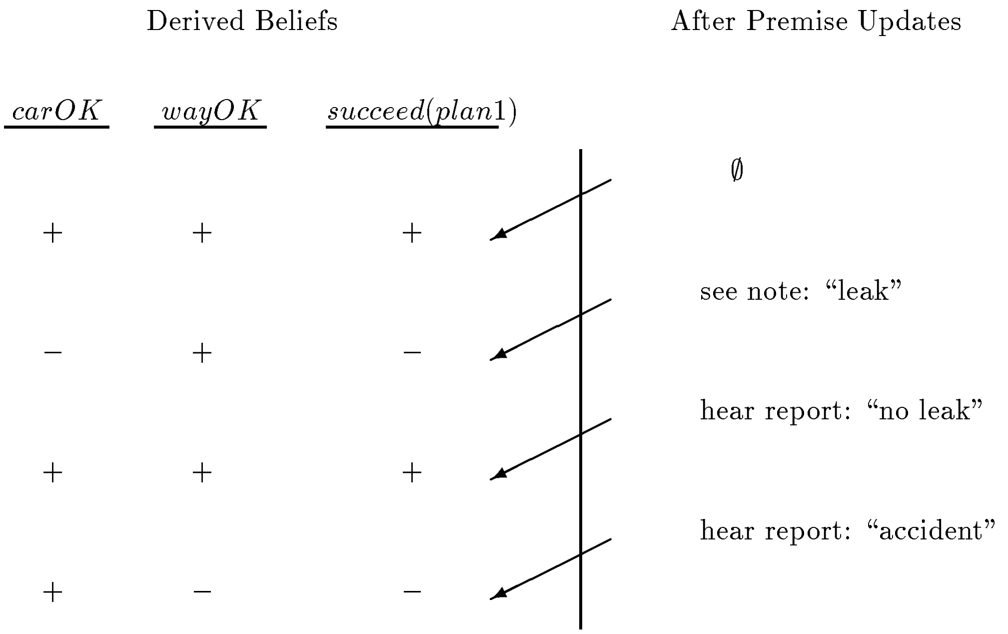


Figure 1.1: Dynamic Evolution of Beliefs: Commuting. Each row of the table (on the left hand side of the figure) indicates the status (+ for "believed" and − for "not believed") of Jon's key beliefs after the premise updates (on the right hand side of the figure) above that row. $\emptyset$ stands for the empty set: no updates received yet. All beliefs and update premises are about *today*.

This story illustrates how practical reasoning often involves the logically non-monotonic, dynamic evolution of beliefs. New information is arriving continually, and one has to revise one's tentative, decision-oriented conclusions in an ongoing way.

Figure 1.1 illustrates the dynamics of the derived beliefs in the story above. For the sake of brevity and simplicity, it focusses on only some of Jon's beliefs.

The usefulness of a logical formalism for default reasoning is that (sometimes) it enables one to represent such reasoning declaratively. The dynamically evolving pattern of belief emerges from a monotonically accumulating set of premise axioms, plus domain-independent logical principles of inference. Thus, for the above story, we can represent Jon's starting knowledge base, and his premise updates, as axioms.[12]

Let Jon's starting premises $\mathcal{A}_0$ consist of: [13]

$$(d1) :> \ carOK(t)$$

$$(d2) :> \ wayOK(t)$$

$$(d3) :> \ carOK(t) \wedge wayOK(t) \ \supset \ succeed(plan1, t)$$

$$\bullet\!> \ \forall t.\, leak(t) \ \supset \ \neg carOK(t)$$

[12] in the Circumscriptive Logic of Defaults, defined in section 3.2

[13] Again, here, we focus on only some of Jon's beliefs.

$\bullet\!> \ \forall t.\ accident(t) \supset \neg wayOK(t)$

where t stands for time.

Let Jon's additionally-learned premises be as follows.
The first update $\mathcal{U}_1$ (from the note) is:

$$(d4) :> \ leak(today)$$

$$\mathcal{PREFER}(d4, d1)$$

The second update $\mathcal{U}_2$ (from the phone call) is:

$$(d5) :> \ \neg leak(today)$$

$$\mathcal{PREFER}(d5, d4)$$

The third update $\mathcal{U}_3$ (from the radio report) is:

$$(d6) :> \ accident(today)$$

$$\mathcal{PREFER}(d6, d2)$$

Then[14] the pattern of belief change in the story corresponds to the conclusions in the non-monotonic logical system:

$$\mathcal{A}_0 \ \mid\!\approx \ carOK(today) \wedge wayOK(today) \wedge succeed(plan1, today)$$

$$\mathcal{A}_0, \mathcal{U}_1 \ \mid\!\approx \ \neg carOK(today) \wedge wayOK(today)$$

$$\mathcal{A}_0, \mathcal{U}_1 \ \not\mid\!\approx \ succeed(plan1, today)$$

$$\mathcal{A}_0, \mathcal{U}_2 \ \mid\!\approx \ carOK(today) \wedge wayOK(today) \wedge succeed(plan1, today)$$

$$\mathcal{A}_0, \mathcal{U}_3 \ \mid\!\approx \ carOK(today) \wedge \neg wayOK(today)$$

$$\mathcal{A}_0, \mathcal{U}_3 \ \not\mid\!\approx \ succeed(plan1, today)$$

Of course, much of the information in the story was not explicitly represented in the above formulation. Some of this was irrelevant to the main commuting problem: for example, that Jon was eating breakfast when he heard the radio report. However, some of this information was about the source of the updates; for example:

- that Jon treats a note or report from his wife as credible, and as taking precedence over his initial default about $carOK$;

[14] Using the Circumscriptive Logic of Defaults, defined in chapter 3. Transitivity and irreflexivity of the precedence ordering are enforced in the formalism.

- that the report from his wife takes precedence over the note from his wife, since it was fresher information from the same source;
- that Jon treats the radio report as credible, and as taking precedence over his initial default about $wayOK$.

Above, we also ignored a number of subtleties; for example

- the (default-type) "frame assumptions" (temporal persistence or inertia) involved in presuming that the past information in the note, the phone report, and the radio report each applied (continued, persisted) to the future situation in which the car or route would actually be used;
- the details of the note (e.g., Jon recognizing that it is from his wife, understanding the writing and the English), and, similarly, of the phone call and of the radio report.
- that the beliefs are those of one agent, Jon, among many.

In addition, we ignored the issue of how inference was controlled, i.e., which inferences were performed, and when.

To give a relatively complete (explicit) account of the kind of default reasoning described in the story, is a challenge that tests the limits of current approaches in AI knowledge representation (KR). Time, action, communication, as well as commonsense and particular-domain knowledge, are all involved in the reasoning: each of these issues occupies a distinct KR research sub-community. Part of our ambition is to advance the understanding of default reasoning to the point where this kind of problem can be handled well.

1.6.3 The Need for Expressive Generality

We are concerned with the need for expressive generality in applications: particularly, in the realm of empiricism and action. Especially, in three regards. Firstly, we observe that many potential applications appear to need the ability to express fairly **arbitrary first-order forms** of beliefs. For example, in scientific theory formation, communication, natural language, and law, one needs to express default beliefs, as well as for-sure beliefs, of fairly arbitrary first-order-logical form. Circumscription, we show after formulating its idea of a default axiom in chapter 3, is able to fulfill this requirement. By contrast, many non-monotonic formalisms are limited to Horn-clause or propositional forms.

Secondly, we find that in order to specify the kinds of preferences among defaults that appear to arise naturally in many applications, that it is important to be able to express fairly **arbitrary partial orders of precedence**. Precedence based on the specificity dominance principle employed, e.g., in default inheritance systems [Touretzky, 1986] [Geffner, 1992] and argument systems [Pollock, 1987] [Loui, 1987], can result in a fairly arbitrary partial order structure. The same goes, we have found, for precedence based on source reliability. For example, in our formulation of the

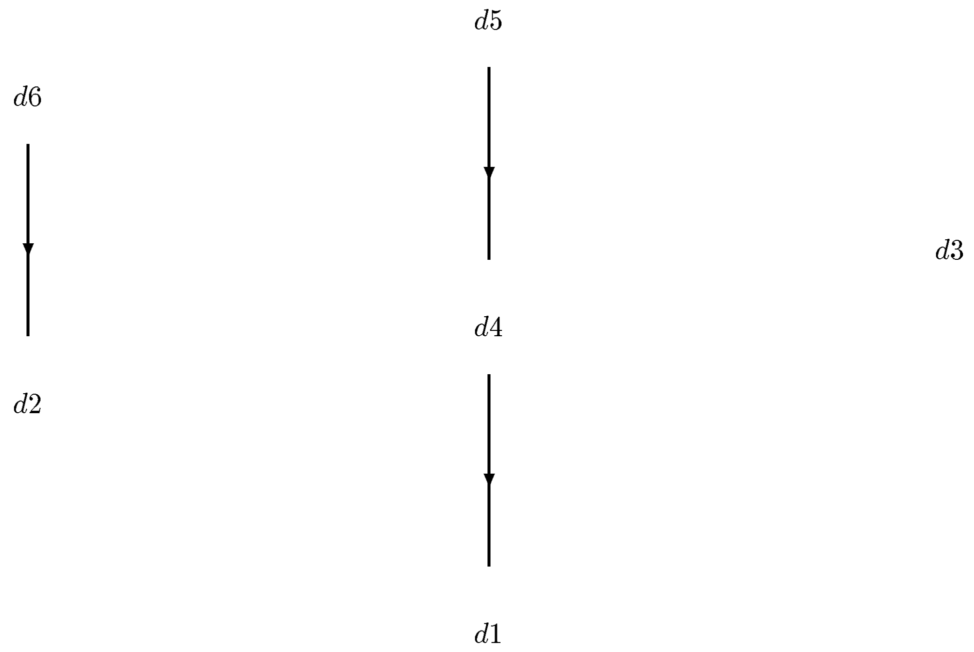


Figure 1.2: "Columnar" precedence among the defaults in the commuting story. Here, a strict partial order is represented as a directed acyclic graph. Each default corresponds to a node. Each directed arc indicates that the "from" node has strictly higher precedence than the "to" node. ($\{d3\}$ forms a degenerate column containing only one node.)

commuting story in section 1.6.2, the precedence partial order did not form a single totally-ordered series of "strata", as studied, e.g., in [Lifschitz, 1985] [Lifschitz, 1987a] [Brown and Shoham, 1989] [Przymusinski, 1988], but rather formed several disconnected "columns" (see Figure 1.2). We will see more examples of precedence in chapter 2.

We find, furthermore, that it is important to be able to **explicitly specify precedence**, e.g., that associated with source reliability. For example, in our formulation of the commuting story in section 1.6.2, the precedence partial order among the defaults was part of the information incrementally acquired via communication, so it was desirable to be able to represent it via a collection of explicit axioms.

Circumscription, after we generalize its idea of prioritization in chapter 2, and its idea of axioms in chapter 3, is able to fulfill both of these requirements for expressing precedence: explicitness, and arbitrary well-founded (e.g., finite) partial orders. By contrast, most non-monotonic formalisms cannot represent precedence at all, can do so only implicitly, or can only represent a limited class of (the finite precedence) partial orders.

Our third observation is about expressive generality in inference, rather than in specification. We find that it is important to treat **updating with new defaults** as a central mode of reasoning for applications in the realm of empiricism and action. Almost all human empirical knowledge is not absolutely for-sure: it is defeasible to some degree. As reasoning agents, humans are continually acquiring such knowledge, e.g., via interpreted sensory-perceptual observations and via communication. We are interested, accordingly, in creating automatic programs that are **learning agents** in this sense of continually acquiring new premise beliefs, including new defeasible premise beliefs. By contrast, the spirit of much, perhaps most, previous work on expressively rich non-monotonic reasoning treats the defaults as a kind of fixed background, and focuses on updates consisting of for-sure information. This is especially true for the work in the realm of representational conventions, but has not been restricted to that. E.g., the articles about the Closed World Assumption and its generalized forms in which all predicates are minimized, and about the semantics of logic programs with negation, in [Ginsberg, 1987]. E.g., [Touretzky, 1986] about inheritance systems. E.g., [Geffner, 1992] [Kraus *et al.*, 1990] [Lehmann and Magidor, 1988] [Lehmann, 1989] about conditional entailment. Little previous work has addressed updating with new defaults as a main concern for inference.[15]

In this thesis, we will be focussing considerably on updating with new default axioms (and their precedences). Interestingly, we will discover that, in our circumscriptive formalism, updating with new default axioms often subsumes, as a special case, updating with new for-sure axioms (see Theorem 5.75 in sub-section 5.13.3, and our results throughout chapter 6).

1.7 Circumscription As A Vehicle

Our investigations of non-monotonic reasoning in this dissertation are primarily in one particular NM formalism: circumscription.

Why Choose the Circumscriptive Formalism

We choose strategically the circumscriptive formalism as our vehicle of analysis for several reasons. Firstly, we are motivated by the need for certain kinds of expressive generality in applications, as we discussed in sub-section 1.6.3. Circumscription is relatively powerful expressively in a way that matches well our main concerns. It can express defaults (as well as for-sure beliefs) of arbitrary first-order-formula form. Especially important for us, it can express precedence: via the formal mechanism of priority. We will show, indeed, how it can express arbitrary finite partial orders of precedence[16], once we definitionally generalize the mechanism of prioritization.[17]. We are less interested in the expressive subtleties, such as treatment of antecedent versus consequent in default rules, or of equality, or of introspection, offered by other formalisms, e.g., Default Logic and Autoepistemic Logic, that are not offered by circumscription. More generally, circumscription is able to represent fairly general kinds of model preference criteria (cf. [Shoham, 1988] [Brown and Shoham, 1989]) as sentences within second-order classical logic.

[15]One of the few results in this area is the "semimonotonicity" result (Theorem 6.3) in [Reiter, 1980].

[16]and, more generally, arbitrary well-founded infinite partial orders of precedence

[17]See discussion of other formalisms' capabilities for expressing precedence in sections 2.11 and 8.5 through 8.8.

Secondly, circumscription is relatively nice to work with mathematically. It was, when we embarked on our project, and still is, probably, the best-explored of the more expressive non-monotonic logical systems. It has a classical, Tarskian semantics. Unlike many NM formalisms, e.g. Default Logic and Autoepistemic Logic, it is skeptical rather than credulous: a single body of conclusions, rather than several (possibly infinitely many) mutually exclusive alternatives, is sanctioned. (See below for more discussion of skepticality.) Therefore, one can neatly axiomatize an entire NM theory (or sub-theory) in classical logic[18] as a single expression. Overall, a major advantage is that the classical tautologies and inference rules are relatively directly applicable for proving properties about circumscriptive theories, particularly about their decompositions.

Thirdly, there do exist some (backward) proof procedures for fairly general cases of circumscription [Przymusinski, 1989] [Ginsberg, 1989] [Baker and Ginsberg, 1989] [Inoue and Helft, 1990] [Helft *et al.*, 1991], which makes it relatively "for real" as a prospective testbed in the near future for our results about inference and applications. (See section 7.4 for discussion of these procedures.)

Skeptical Sí! Credulous No!:
Circumscription, unlike many other NM formalisms (e.g., Default Logic and Autoepistemic Logic; see sections 8.5 and 8.6), is skeptical: a single body of conclusions, rather than several mutually exclusive alternatives, is sanctioned. (Recall the terminology of "skeptical" versus "credulous" from page 12.) We believe that skepticality of the NM formalism is more appropriate for our intended purposes than credulousness. In classical logic, the main point of the concept of entailment or logical implication from some given premise axioms is that the entailed sentences are those which are beliefs sanctioned in *all* legitimate scenarios permitted by the premises. These scenarios are formalized as interpretations or models in classical logic and in some skeptical NM formalisms, e.g., circumscription; and as "extensions" (as in the phrase "multiple extensions") in many credulous NM formalisms, e.g., Default Logic and Autoepistemic Logic. Yet if you have to act or decide on your beliefs, e.g., as in our intended applications of learning agents, then you want stuff you really and most believe, not a fanciful panoply of speculative or conjectural "extensions" as in the credulous formalisms. Moreover, though one can define skeptical versions of most credulous NM formalisms, circumscription in particular mathematically and conceptually facilitates working with a single body of conclusions (see remarks above).

Circumscription's Relation to Other Formalisms

More and more is becoming known about the commonalities between different non-monotonic logical systems. There are now many known relationships (via transitivity) between circumscription and several other formalisms, including: Default Logic, Autoepistemic Logic, Reason Maintenance Systems, and inheritance systems. See sections 2.11 and 8.5 through 8.7 for more discussion of how our results bear on other NM formalisms, and of their known inter-relationships.

[18] in second-order logic for most of the theories we will be interested in

Chapter 2

Generalizing Prioritization

In this chapter[1], we generalize the formal idea of prioritization in circumscription, in two ways. The first is to permit the precedence partial order to be any well-founded, e.g. any finite, partial order. Previously, precedence was restricted to be "layered" ("stratified") in a fashion similar to military rank; yet our intended applications demand it be non-layered. The second is to enable precedence and prioritization to be defined over groups of defaults, in hierarchical fashion. We go further and show how prioritization can be generalized to apply even beyond circumscription, to the aggregation of arbitrary preference criteria, whether in the realm of NM reasoning or not, as long as the criteria are transitive and reflexive. We show that our idea of prioritization subsumes several previous ideas of precedence in the NMR literature. Along the way, we review and discuss circumscription in rather gory technical detail: in preparation for its use in the rest of the thesis.

Guide to Reader: Section 2.12 provides a summary / overview of the whole chapter.

2.1 Introduction

The concept of prioritization in circumscription, and related ideas in other non-monotonic formalisms, have received much attention in the AI community recently. There are several intuitions behind the concept of prioritization. Prioritization can be viewed as a tool to specify *confidence* information about default beliefs. Alternatively, it can be viewed as a tool to specify semantic, or inferential, *precedence*. However, a full discussion of the motivations and applications of the idea of prioritization should be a subject of (at least) a whole paper in itself. Rather than repeating the justifications for interest in prioritization, here we will mainly take it as a given. In this chapter, we *identify several bases for prioritization information.* This motivates us to generalize the *formal* idea of prioritization as a method to aggregate preferences, so that we can then apply it to define new, more expressively general variants of the circumscription formalism.

Terminology: Prioritization Vs. Precedence:

In chapter 1, we spoke of the partial order of precedence. Prioritization is one particular (precisely

[1] revised and expanded from [Grosof, 1991]

defined) representation or interpretation of the idea of precedence.[2] Henceforth, by a "prioritization partial order", we mean a precedence partial order in that particular context, e.g., in the context of prioritized circumscription.

The concept of prioritization in circumscription was introduced by John McCarthy in [McCarthy, 1986]. However, he did not give a technically precise definition of prioritization; rather, he gave some examples, and suggested considering partial orders of priorities. Vladimir Lifschitz in [Lifschitz, 1985] defined "prioritized circumscription" in detail. However, his definition imposes some restrictions that we show makes it unsuitable for many applications of default reasoning, including inheritance. Firstly, the structure of the prioritization partial order (i.e., the strict partial order over the set of minimized predicates) was restricted there to be what we call *layered* (what some others have called "stratified"). Yet the kind of confidence, or *precedence*, that arises naturally in applications, for example, from the specificity dominance principle in inheritance, is often *not* layered. Here, we overcome the previous limitations by showing how to generalize the prioritization partial order to be *any well-founded* (e.g., finite) strict partial order, including *non*-layered.

Secondly, both McCarthy and Lifschitz defined prioritization only in the context of minimizing *individual predicates* in circumscription. We show how to *prioritize arbitrary model-preference criteria, and their syntactic correspondents.* One application of this is to generalize prioritized circumscription. We also develop a concept of *composing* prioritization and show how to prioritize preference criteria expressing internally-prioritized *groups* of minimized predicates (or defaults). We show that this enables one to express hierarchical modularity in the specification of default reasoning. Our definition of prioritization is applicable to any kind of preference criteria, even those not about models, as long as they obey the properties of reflexivity and transitivity.

We discuss how our generalized prioritization relates to similar ideas in non-monotonic formal systems other than circumscription. We show that it unifies and subsumes as special cases several such ideas of precedence. We give further justification to our definition by showing that it satisfies several intuitive desiderata, including those of [Doyle and Wellman, 1991].

2.2 Pre-Orders and the General Definition of Circumscription

Note to the Reader:
In this section and the next, we review circumscription and discuss some technicalities revolving about it that we will need later. Even if you are familiar with circumscription, you should at least skim these sections (2.2 and 2.3), since some of the definitions, assumptions, observations, and theorems are new and/or particular to us.

Our review of circumscription is by no means comprehensive. To familiarize yourself with the basic historical development of circumscription, the most important papers to read are: [McCarthy,

[2] See sections 2.11 and 8.3 for discussion of its interpretations in some other NM formalisms.

1980] [McCarthy, 1986] [Lifschitz, 1984] [Lifschitz, 1985]. For a textbook introduction to circumscription, see [Genesereth and Nilsson, 1987] (sections 6.4 and 6.5).

General circumscription was defined by [Lifschitz, 1984] as the minimization of a preference criterion that has the property of being a *pre-order*.

Definition 2.1 (Pre-Order)
A binary relation H is a *pre-order* when it is reflexive and transitive.

$$\forall x.\, H(x,x)$$
$$\forall x,y,z.\, H(x,y) \wedge H(y,z) \supset H(x,z)\ .$$

A (non-strict) partial order, by contrast, is a pre-order that also obeys "anti-symmetry", i.e.:

$$\forall x,y.\, H(x,y) \wedge H(y,x) \supset x = y\ .$$

Example 2.2 (Pre-Order: Card-Game Versatility)
An example of a pre-order G is: "at least as versatile at playing card games", in the following sense. Let the domain be the set of all people. Let the set of card games be:

$$\{poker, bridge, gin, pinochle, crazy8, solitaire, war\}$$

We define one person, say $Julie$, to be at least as versatile as another person, say Tim, at playing card games if $Julie$ knows the rules (and basic strategy) for every card game that Tim knows. Then G is a pre-order: it is reflexive and transitive. G is not total: two persons might know disjoint sets of games, so that neither is at-least-as-versatile as the other. E.g., it might be that $Fred$ knows only $bridge$ and $poker$, while Tim knows only $pinochle$ and $solitaire$. (We then say that $Fred$ and Tim are incomparable with respect to G.) Since two different people, say $Julie$ and $Janine$, might know the same games, G is not quite a partial order: it lacks anti-symmetry. ($Julie$ and $Janine$ then lie in an equivalence class with respect to G.)

Notation and Terminology for Pre-Orders:
We adopt the following notation for pre-orders. We write $x \preceq_H y$ to stand for $H(x,y)$, $x \approx_H y$ to stand for $(x \preceq_H y) \wedge (y \preceq_H x)$, and $x \prec_H y$ to stand for $(x \preceq_H y) \wedge \neg(x \approx_H y)$. Note that the lack of the anti-symmetry property means that, in general, $x \approx_H y$ does not imply $x = y$.

We say that a pre-order is *defined over a domain* $\mathcal{D}$ when both of its arguments are drawn from that domain.

Sometimes, we will show the arguments of a pre-order explicitly by using lambda notation; especially when giving the definition of a pre-order. E.g., we may write H as $\lambda Z, Z'.\, H(Z, Z')$.

Pre-orders are also sometimes called quasi-orders.

Intuition about Pre-Orders: Equivalence Classes:
One can view a pre-order as a strict partial order on equivalence classes, where the equivalence is

defined by the $\approx_H$ relation. (See Fact B.3 in Appendix section B.2 for a formal statement of this point.)

Intuition about Pre-Orders: Voting:

Viewing a pre-order as a preference criterion or "voting function", transitivity seems a fairly basic desirable property. Reflexivity means that the preference is non-strict. Lack of anti-symmetry, as well as lack of totality, means that the criterion is quite *partial* in a certain sense: there is a class of comparisons for which the "voter" says "it's a tie", and another for which she abstains ("I don't have anything to say").

Definition 2.3 ("Base" Language; Pre-Orders in Circumscription)

In the context of circumscription, we will be discussing pre-orders that are defined over predicate and function symbols of a *"base" logical language* $\mathcal{L}$. We will define pre-orders, in classical second-order logic, that take as explicit arguments a tuple of predicate and function symbols "similar to", i.e. of the same type as, the tuple of all predicate and function symbols in $\mathcal{L}$ (see Z notation below). (**Terminology: Symbols**: We will use the word "symbols" to mean non-logical symbols, i.e., predicate and function symbols, throughout this thesis.) Mostly, we will be talking about circumscriptions representing prioritized default theories. In that context, $\mathcal{L}$ is the logical language used to express the formula forms of the for-sure and default beliefs. In all of our examples and discussions about circumscriptive default theories, we will be presuming (unless stated explicitly otherwise) that the form of those beliefs are classical first-order. Thus, in that context, $\mathcal{L}$ will be first-order. More generally, though, we will permit $\mathcal{L}$ to be second-order. [3] Nothing inherent about the idea of circumscription, however, prevents considering other kinds of "base" languages: e.g., classical higher-order, or modal. [4]

Assumption: Base Language and Axiomatizations are Finite:

We **assume** throughout this thesis, by convention, that the base language $\mathcal{L}$ is finite. That is, we assume it has only a finite number of predicate and function symbols; and we consider only finitely write-able formulas. This assumption is usual in AI Knowledge Representation: one works with specifications of theories; these specifications contain only a finite number (and size) of axioms.

Why Second-Order?:

Technically, with a (propositional or) first-order "base" language for expressing for-sure and default axioms, the useful pre-orders will be defined in second-order logic, rather than first-order logic. This is because they represent preference criteria for comparing one theory (or model) defined over $\mathcal{L}$, with another theory (or model), also defined over $\mathcal{L}$. Intuitively, second-order is needed to talk, all at the same time, about the multiple, different theories or models. In the context of prioritized default

[3] We will sometimes need to consider second-order: for example, when working with serial decompositions of circumscriptions (defined in section 4.5).

[4] See definition of "prioritized logic" in subsection 2.11.2, and Lifschitz' [1990a] examples of higher-order circumscriptive bases, for some discussion of how.

theories, the second-order form of the pre-orders will be fairly simple: not involving second-order quantifiers, only second-order variables.

Definition 2.4 (Similar)
We define two predicates to be *similar* when they have the same arity. Ditto for two functions. More generally, two open formulas are similar if they have the same arity of free variables. Ditto for two terms. We can view a predicate as an open literal, and a function as an open term.

Notation:
Let $D \leq E$ stand for $\forall x.\ D(x) \supset E(x)$, where D and E are open formulas that are similar, and x stands for a tuple of individual object variables. Let $D{=}E$ and $D < E$ then be respectively defined, analogously to $\approx_H$ and $\prec_H$ above, as $(D \leq E) \wedge (E \leq D)$ and $(D \leq E) \wedge \neg(D{=}E)$. We also apply this notation to tuples $D = \langle D_1, \ldots, D_m \rangle$ and $E = \langle E_1, \ldots, E_m \rangle$, assuming that they are similar: $D \ \leq \ E$ stands for

$$D_1 \leq E_1 \ \wedge \ \ldots \ \wedge \ D_m \leq E_m \ .$$

Terminology and Notation: Domains of Pre-Orders for Circumscription:
We will say that a pre-order H *is defined over* a domain of tuples similar to a tuple of symbols Z when H is the binary relation

$$\lambda Z^1, Z^2.\ H(Z^1, Z^2)$$

, where Z^1 and Z^2 are similar to Z, and the two properties

$$\models \quad \forall X.\ H(X, X)$$

$$\models \quad \forall X, Y, Z.\ H(X, Y) \wedge H(Y, Z) \ \supset \ H(H, Z)$$

hold. (See example of lambda notation below.)

When a reference tuple Z of variables is clear in context, we will often say simply that the pre-order H is defined over Z, meaning that its two arguments must both be similar to Z. Note that the lack of the anti-symmetry property lets us regard any pre-order defined over Z as **implicitly also defined over any bigger tuple W of symbols** that includes Z as a sub-tuple.[5] (we give an example of this below). Thus often we will omit the explicit λ part even when defining pre-orders.

Notation: we will usually, and by default, use Z to denote the tuple of all symbols in the base language $\mathcal{L}$.

Example 2.5 (Defining the Domain of a Pre-Order)
As an example of a pre-order being implicitly defined over a bigger tuple of symbols: consider the following. Let Y be the tuple

$$\langle adult, works \rangle$$

of unary (1-ary) predicate symbols. Let G be a pre-order, defined over Y, i.e., over the domain of tuples similar to Y, where:

[5] For example, with any tuple of fixed symbols adjoined to Z.

$$\langle adult^1, works^1\rangle \preceq_G \langle adult^2, works^2\rangle \stackrel{\text{def}}{\equiv}$$
$$[\forall x.\ (adult^1(x) \supset works^1(x)) \ \supset\ (adult^2(x) \supset works^2(x))]$$

That is, here, $\langle adult^2, works^2\rangle$, i.e., Y^2, is non-strictly preferred to $\langle adult^1, works^1\rangle$, i.e., Y^1, iff the formula $adult^2(x) \supset works^2(x)$ subsumes the formula $adult^1(x) \supset works^1(x)$.

Intuitively, maximizing this preference corresponds to the default that adults work. (By "preference" we mean with respect to G.) We will discuss this more in Definition 2.11.

Equivalently, we can state the above definition of G in terms of **lambda notation**. We say that G is defined to be the relation:

$$\lambda Y^1, Y^2.\ [\forall x.\ (adult^1(x) \supset works^1(x)) \ \supset\ (adult^2(x) \supset works^2(x))]$$

Let Q be the tuple

$\langle adults, works, student, studies\rangle$

of unary predicate symbols. Q is thus a bigger (in the sense of superset) tuple of symbols than Y. Then G is *implicitly* defined over Q, i.e., over the domain of tuples similar to Q. That is,

$$\langle adult^1, works^1, student^1, studies^1\rangle \preceq_G \langle adult^2, works^2, student^2, studies^2\rangle \stackrel{\text{def}}{\equiv}$$
$$[\forall x.\ (adult^1(x) \supset works^1(x)) \ \supset\ (adult^2(x) \supset works^2(x))]$$

Re-stating this in terms of lambda notation: G is defined to be the relation:

$$\lambda Q^1, Q^2.\ [\forall x.\ (adult^1(x) \supset works^1(x)) \ \supset\ (adult^2(x) \supset works^2(x))]$$

In like fashion, G is implicitly defined over any W that includes Y as a sub-tuple.

Definition 2.6 (Predicate Pre-Order)

The focus of most previous work on circumscription has been on minimizing predicates. Let $P \in Z$ be a single predicate in $\mathcal{L}$. We define the individual *predicate pre-order* corresponding to P as:

$\lambda Z^1, Z^2.\ P^1 \leq P^2$

, which we will sometimes write as $\preceq_P$. For example, if $flies, ab \in Z$, then

$\lambda Z^1, Z^2.\ flies^1 \leq flies^2$

and

$\lambda Z^1, Z^2.\ ab^1 \leq ab^2$

are each an individual predicate pre-order.

Intuitively, one can view minimizing a predicate pre-order as a preference for the predicate to have a smaller extension, where by "smaller", we mean in the sense of subset. Minimizing a predicate pre-order in circumscription means trying to make the predicate be true for fewer individuals (fewer in the sense of subset).

Definition 2.7 (General Circumscription)

Circumscription in its general form was defined by [Lifschitz, 1984] as minimization with respect to a pre-order. Let $B[Z]$ be a formula in $\mathcal{L}$, where Z is the tuple of all predicate and function symbols[6] in $\mathcal{L}$. (It will be helpful usually to think of B as a closed formula, i.e., a sentence.) Let H be a pre-order defined over Z. The circumscription of (Z in) $B[Z]$ with respect to H is the second-order[7] formula

$$C(B; H; Z) \stackrel{\text{def}}{\equiv}$$

$$B[Z] \wedge \neg \exists Z'.\, B[Z'] \wedge \ Z' \prec_H Z$$

Terminology: We call $B[Z]$ the **base** sentence, H the **minimization pre-order**, or **preference pre-order**, and Z the **varied** predicates and functions, of the circumscription.[8] We will, also, sometimes call the circumscription "**the circumscription formula**" or the "**circumscription sentence**".

Terminology and Notation: The Augmentation Part of a Circumscription:
We refer to the quantified part of the circumscription formula as its **augmentation** or augmentation part. We denote it with $C_{aug}(B; H; Z)$. Intuitively, the augmentation corresponds to the conclusions that are sanctioned non-monotonically by the circumscription, but are not sanctioned by the base sentence $B[Z]$ alone.

Definition 2.8 (Policy)
We define the *policy* of a circumscription as the part of its specification other than its base sentence. Such a specification might include prioritization and the "fixing" of some symbols. However, later we will show that any (such) policy can be viewed simply as the specification of the one overall minimized pre-order in general circumscription.[9]

Definition 2.9 (Predicate Circumscription)
We define *predicate circumscription* as circumscription with respect to a predicate pre-order.

Example 2.10 (Predicate Circumscription)
Let B be defined as the conjunction of the sentences:

$$flies(a)$$

$$flies(b) \vee flies(c)$$

---

[6]However, we will mainly be considering the case excluding functions: see page 36.

[7]More precisely, the circumscription will be a second-order formula if $\mathcal{L}$, B and H are second-order, as discussed in Definition 2.3.

It is straightforward to generalize circumscription to higher-order logic, beyond second-order logic, and also to other logics, but we will not take the trouble.

[8]See sub-section 2.3.1 for a discussion of "fixing", i.e., not varying, some of the predicate and function symbols. We show there that that can be accomplished within the above definition, in which technically all symbols are varied.

[9]For discussion of "fixing", see sub-section 2.3.1, especially Theorem 2.19, sub-section 2.3.2, and section 3.5.

together with the Uniqueness of Names Axiom

$$(a \neq b) \wedge (a \neq c) \wedge (b \neq c)$$

which, for brevity, we will denote by $UNA[a,b,c]$. (See Definition 2.21, and our assumption in examples, about uniqueness of names, in section 2.3.) Let H be the predicate pre-order corresponding to $flies$. Then the predicate circumscription $C(B;H;Z)$, in which the predicate $flies$ is minimized, is:

$$\begin{aligned} & flies(a) \wedge (flies(b) \vee flies(c)) \wedge UNA[a,b,c] \\ & \quad \wedge\ \neg\exists flies', a', b', c'.\ \{flies'(a) \wedge (flies'(b) \vee flies'(c)) \\ & \qquad\qquad\qquad\qquad\qquad \wedge\, UNA[a,b,c] \wedge (flies' < flies)\} \end{aligned}$$

This, it turns out (using theorems in [Lifschitz, 1985]), is equivalent to:

$$\begin{aligned} B[Z] \wedge \{ & (\forall x.\ flies(x) \ \equiv\ ((x=a) \vee (x=b))) \\ & \vee (\forall x.\ flies(x) \ \equiv\ ((x=a) \vee (x=c)))\} \end{aligned}$$

In English: the circumscription, due to its augmentation part, sanctions exactly the additional conclusion that the set of fliers consists either of only $\{a,b\}$ or of only $\{a,c\}$. Thus, for example,

$$C(B;H;Z) \ \models\ \neg flies(b) \vee \neg flies(c)$$

Definition 2.11 (Default Pre-Order)

A second interesting kind of pre-order is what we call a *default pre-order*, or *elementary formula pre-order*. Let $\lambda x.\ E[Z,x]$ be an open (or closed) first-order formula of $\mathcal{L}$ in predicates (tuple) Z, where x is a tuple of object variables. Then we define the corresponding (individual) elementary formula pre-order, i.e., default pre-order, as $E[Z] \leq E[Z']$, e.g.,

$$\lambda Z, Z'.\ (bird \supset flies) \leq (bird' \supset flies')$$

where $bird, flies \in Z$. Intuitively, in the context of circumscription, maximizing a default pre-order corresponds to the preference or "attempt" to make the default formula be true for as many of its instances "as possible": e.g., for the default pre-order above, to make as many objects x "as possible" have the property that

$$bird(x) \supset flies(x)$$

Model-theoretically, a default pre-order $\preceq_E$ represents the preference for a larger (in the sense of superset) extension of the default formula E. (see sub-section 2.2.1 below for more about models). Recall our intuitive introduction of the concept of a default on page 11. There, we spoke in terms of consistency. In the context of circumscription, consistency with the for-sure-status premise beliefs constrains the maximizing activity of a default pre-order.

McCarthy [1986] **defined "formula circumscription"** as the minimization of an individual default pre-order. [10] An individual predicate pre-order is the special case of an individual default pre-order where the formula $E[Z, x]$ is just an open literal $P(x)$. Minimizing the formula $\neg E[Z]$ is just equivalent to *maximizing* the formula $E[Z]$.

Definition 2.12 (Default Circumscription)
We define *default circumscription* as circumscription that maximizes a default pre-order. (See Definition 2.43 for more details.)

2.2.1 Models and Theories

Preferences On Models:
The essence of the definition of general circumscription is that the *pre-order represents a preference criterion*. For the particular kinds of pre-orders that we will be discussing in detail in this thesis (i.e., prioritized predicate and default pre-orders[11]), the preference pre-order formula $\preceq_H$, defined on tuples Z of all predicate and function *symbols* corresponds equivalently to a (reflexive, transitive) preference relation $\sqsubseteq_H$ defined on *models* M. By "equivalent correspondence" here, we mean that the models of $C(B; H; Z)$ are exactly those models of $B[Z]$ that are most preferred with respect to $\sqsubseteq_H$. In reading such a circumscription, one can view a symbol tuple Z as a syntactic "stand-in" for a model M.

Notation:
We will use $\sqsubseteq_H$, $\sqsubset_H$, and $\cong_H$ to stand for the correspondents of $\preceq_H$,$\prec_H$, and $\approx_H$, respectively.

Definition 2.13 (Circumscriptive Theory)
The *circumscriptive theory* is defined as the set of sentences satisfied in all (standard, classical) models of $C(B; H; Z)$, i.e., in all most preferred models of B. This style of definition follows [Lifschitz, 1988a]. [12]

For prioritized predicate and default pre-orders (see mention above, and sub-section 2.7.1), the mapping from a syntactically defined pre-order to its model-preference correspondent is straightforward (and invertible): essentially, one replaces a predicate or function symbol by its denotation,

[10] The reason for "elementary" in our terminology is that many other kinds of pre-orders, e.g., *prioritized*-default pre-orders (defined in sub-section 2.7.1) can also be written as formulas, albeit more complex ones.

[11] See sub-section 2.7.1 for their definitions.

[12] As he observes, the incompleteness of derivability relative to entailment in second-order logic (and higher-order logic) (for standard models) implies it is desirable to avoid a general definition of the circumscriptive theory as the consequential closure of the circumscription sentence. We remark, however, that a closure-style definition would be OK for "collapsible" cases in which the circumscription sentence is equivalent to a first-order sentence, since first-order derivability is complete. Etherington [1988] (chapter 6) suggests that one might, alternatively, use the weaker, non-standard notion of models for second-order (and higher-order) logic. This weak version of second-order is then equivalent to first-order.

and one interprets quantified implication as a subset-inclusion relationhip. See [Etherington, 1988] (chapter 6) for details. In general, however, as Etherington remarks there, the existence of an equivalent model-preference correspondent to any particular syntactically-defined pre-order is an open question.

Assumption: Model-Preference Correspondence:

For the kinds of pre-orders and circumscriptions whose circumscriptive theories we are concerned with in this thesis, we **assume** that:

- The preference pre-order formula $\preceq_H$, defined on tuples Z of all predicate and function *symbols* does correspond equivalently to some (reflexive, transitive) preference relation $\sqsubseteq_H$ defined on *models* M.
- The models of $C(B; H; Z)$ are exactly those models of $B(Z)$ that are most preferred with respect to $\sqsubseteq_H$.

Terminology: Circumscription Vs. Circumscriptive Theory:

To save breath, as a convention, henceforth we will usually identify the circumscriptive theory with the circumscription formula, when the distinction between them is not important. E.g., when discussing our main focus: prioritized default circumscriptions.

Circumscriptive Theory As Non-Monotonic:

A circumscription formula is an expression in classical logic. Thus we can speak of its conclusions in the sense of ordinary classical entailment, which is monotonic: e.g., as we did in Example 2.10. Yet, when we view a circumscription as representing a NM theory, then, in another sense, its conclusions are (in general) non-monotonic: e.g., in Examples 2.25, 2.27, and 2.29.

2.3 More Technical Details About Circumscription

2.3.1 Fixtures

"Fixing" is an expressively important technical aspect in the definition and application of various historical variants of circumscription, e.g., in [McCarthy, 1986] [Lifschitz, 1985]. In order to express desired non-monotonic inferences, it is sometimes useful to exclude some symbols from being varied and (second-order) quantified in the augmentation part of the circumscription formula. We then say that these symbols are "fixed" in (the policy of) the circumscription. Later, we will see some examples of how fixing some symbols affects the circumscriptive theory: it prevents some conclusions. Controlling which symbols are varied, versus which are fixed, is often important in order to achieve the desired set of non-monotonic conclusions generated by a predicate circumscription, for example. The essential intuition behind fixing is **immunity**: that no new non-monotonic conclusions are drawn, beyond the for-sure only information, about sentences involving only fixed symbols. See section 3.5 and sub-section 2.3.2 for more discussion of fixing's effect and of its use for representation in circumscription. In this sub-section, we generalize the idea of fixing and discuss some of its properties. Part of our aim in going through these details is to clarify the relationship between the general definition of circumscription, and historical notations for circumscription that involve fixing.

Next, we define circumscription with some symbols fixed, as a generalization of Definition 2.7. But then we show how it reduces to a special case of Definition 2.7.

Definition 2.14 (Fixed Symbols)
In Definition 2.7, let W be a sub-tuple of Z, and let $Y = Z - W$, so that $Z = \langle W, Y \rangle$. We define the circumscription where, in addition, the symbols W are *fixed*, as:

$$C(B; H; fix\ W; Z) \;\stackrel{\text{def}}{\equiv}\; B[Z] \;\wedge\; \neg\exists Z'.\, W' = W \wedge B[Z'] \wedge Z' \prec_H Z$$

Here $W' = W$ means that each function in W' is everywhere equal to its correspondent in W, and that each predicate in W' is everywhere equivalent to its correspondent in W.
Notation: Implicit Fixture:
Equivalently, we sometimes write the above as:

$$C(B; H; Y) \;\stackrel{\text{def}}{\equiv}\; B[W, Y] \;\wedge\; \neg\exists Y'.\, B[W, Y'] \;\wedge\; \langle W, Y' \rangle \prec_H \langle W, Y \rangle$$

This treats the fixed symbols as more implicit notationally. In general, when writing circumscriptions in this thesis, when we list the explicit variable symbols, there may be other, *implicitly fixed* symbols.

More generally, we define circumscription where an arbitrary equivalence relation, defined over Z, is fixed. An *equivalence relation* is just a pre-order that is *symmetric*:

$$\forall x, y.\, H(x, y) \equiv H(y, x) \qquad ; i.e. :$$

$$\forall x, y.\, x \preceq_H y \equiv x \approx_H y$$

Definition 2.15 (Fixed Equivalence Relation)
Let F be an equivalence relation defined over Z. Then

$$C(B; H; fix\ F; Z) \;\stackrel{\text{def}}{\equiv}\; B[Z] \;\wedge\; \neg\exists Z'.\, B[Z'] \;\wedge\; Z' \approx_F Z \;\wedge\; Z' \prec_H Z$$

Thus fixing a tuple of symbols is just a special case of fixing an equivalence relation: let $Z' \approx_F Z$ be defined as $W' = W$.

Definition 2.16 (Fixing a Pre-Order)
More generally, we will speak of fixing an arbitrary pre-order (not just equivalence relation) $\preceq_G$: by which we mean fixing $\approx_G$. Including in the **notation**: will permit F above to be a pre-order.

Definition 2.17 (Fixing a Formula)

We will also find it useful to speak of *fixing an (elementary) formula*. To fix an open formula $E[Z, x]$, we fix its corresponding elementary formula pre-order.

Note that fixing a formula is expressively reducible to fixing an equivalent predicate: see Proposition 3.34 in sub-section 3.5.3, and sub-section 3.5.4.

Fixing Several Pre-Orders:
Note that fixing a tuple of pre-orders (equivalence relations) can be expressed equivalently as fixing a single pre-order: the conjunction of those pre-orders.[13] Thus multiple fixtures reduce to a single fixture, mathematically.

Observation 2.18 (Fixing Functions Via Fixing Formulas)
One can represent fixing functions as fixing elementary formulas: fixing the function f is equivalent to fixing the elementary formula $\lambda x, y.\ f(x) = y$. See Theorem 2.24 below.

Theorem 2.19 (Expressing Fixture in the Minimized Pre-Order)
Circumscription with fixture of an equivalence relation can be reduced to general circumscription: one can express the fixture as part of the minimized pre-order. Let $\preceq_L$ be defined as $\preceq_H \wedge \approx_F$. Then $\preceq_L$ is a pre-order, and

$$C(B; L; Z) \quad \equiv \quad C(B; H; fix\ F; Z)$$

Proof Overview: Fairly straightforward. See Appendix. **N.B.: READ CHAPTER A of the Appendix BEFORE READING ANY OF THE PROOFS IN THE APPENDIX!!!**
□

Theorem 2.19 shows that we can view the policy of any circumscription with fixed symbols (and more generally, with a fixed equivalence relation) as simply the specification of one pre-order to be minimized while permitting all symbols in $\mathcal{L}$ to vary. That is, any (such) policy can be viewed as simply the specification of the one overall minimized pre-order in general circumscription. [14]

Observation 2.20 (Fixing and Varying Is The Same As Fixing)
Fixing and varying the same symbol is equivalent to fixing it. Fixing should be viewed as a constraint.

More About Fixing: in Chapter 3:
See sub-section 2.3.2 and section 3.5 for much more about fixing.

[13] Alternatively: the parallel prioritization of those pre-orders. (See Definition 2.44.)

[14] A Technical Note: As we will see later (Definition 2.44), conjoining corresponds to a parallel prioritization. Thus control of which predicates and functions are varied, versus which are fixed, is expressible via the minimized pre-order in prioritized predicate circumscription, prioritized default circumscription, and general prioritized circumscription, which we define in sub-section 2.7.1 without explicit mention of fixing.

2.3.2 Functions and Equality

Assumption: All Functions Are Fixed:
To simplify our notation and definitions of circumscriptions, we will assume henceforth throughout this thesis, except where explicitly stated otherwise, that *all function symbols are fixed*, i.e. not varied, in the circumscription, and thus our notation will take this as implicit. We note that both McCarthy's [1986] definition of predicate circumscription, and Lifschitz' [1985] definition of layered-prioritized predicate circumscription have this restriction, as does almost all previous work on circumscription. (One exception is Lifschitz' [1984] definition of general circumscription.)
Dependency of Results on this Assumption:
However, our definitions generalize straightforwardly to include varying and talking explicitly about functions. Many of our results hold when functions are permitted to vary. However, some of our results and discussion depend essentially on the fixture of functions, including:

- some of our sufficient conditions on satisfiability and well-behavior of circumscriptions; and
- all examples, unless mentioned otherwise, with axioms mentioning specific named individuals or terms (see further assumption below of unique names);
- strongly local decomposition, and its implications, for totally-prioritized propositional defaults;
- decompositions of open defaults into default instances.

In addition, some of our results on (definitions of generalized positivity and) monotonicities of updating, involving term substitution and instantiation, assume fixture of functions. However, these can be generalized to the case of variable functions: only terms that are fixed can be substituted or instantiated.

These dependencies will be noted where relevant results are discussed.

Expressive Inessentiality of Functions:
Note that functions are inessential expressively, in the following sense. In $\mathcal{L}$, one can represent functions as (functional) predicates, then include, as an axiom (in the base sentence), the constraint that these predicates are functional.
Assumption in Examples: Uniqueness of Names:
In all of our examples, except where explicitly stated otherwise, we assume the uniqueness of all names.

Definition 2.21 (Uniqueness of Names)
The uniqueness of names is a standard notion in the logical literature in computer science. It means the distinctness, i.e., inequality, of all terms ("names") in the base language $\mathcal{L}$. In first- and higher-order logic, the uniqueness of names is a schema, rather than an axiom, in general: since there may be an infinite number of terms.

In the logical literature, and, especially, in the NMR literature, uniqueness of names is sometimes defined, furthermore, as a logically non-monotonic schema. Typically, a finite collection of explicit equality (and possibly inequality) axioms is given, then a closed world assumption is made on equality. This closed world assumption is non-monotonic, and is often called the Uniqueness of Names Axiom, the Uniqueness of Names Assumption or the Unique Names Hypothesis.

In this thesis, we will not worry about how the distinctness of all terms gets established; we will view uniqueness of names as logically monotonic.

Definition 2.22 (Complete Theory of Equality)
Another standard notion in the logical literature is: completeness of a theory of equality. Let $B[Z]$ be a formula in the base language $\mathcal{L}$. We say that B includes, or implies (entails), *a complete theory of equality* when:

$$\forall x, y.\, (B[Z] \models (x{=}y)) \quad \vee \quad (B[Z] \models (x \neq y)$$

Definition 2.23 (Domain Closure)
In a first-order theory, we say *the domain is closed* when the theory entails that all individuals belong to a finite named set. I.e., we say that the sentence $B[Z]$ satisfies (overall) domain closure when B entails that

$$\forall x.\, (x = t1) \vee (x = t2) \vee \ldots \vee (x = tm)$$

where $t1, \ldots, tm$ are a finite set of ground terms. We call these ground terms: the **domain elements**. We call the set of the domain elements: **the domain**.

Theorem 2.24 (Fixing of Functions)
Some equivalent and sufficient conditions for the fixing of functions are as follows. The first two points are new as far as we are aware. The last two follow easily from the second.

1. Fixing the function f is equivalent to fixing the elementary formula
   $\lambda x, y.\, f(x) = y.$
   Thus, one can represent fixing functions as fixing elementary formulas.

2. Suppose the base B of a circumscription $C(B; H; Z)$ includes a complete theory of equality. Then all function symbols are effectively fixed, i.e., **indirectly** fixed (**terminology**; see Definition 3.47). More precisely: let $F \subset Z$ stand for the tuple of all function symbols. Then
   $C(B; H; fix\ F; Z) \quad \equiv \quad C(B; H; Z)$

3. Domain closure plus a complete theory of equality about the ground terms suffices to indirectly fix (in the sense above) all the functions.

4. Domain closure plus uniqueness of names suffices to indirectly fix (in the sense above) all the functions.

Proof Overview: We show each point from the previous one. See Appendix. □

Note that Ginsberg, for example, assumed a complete theory of equality in his circumscriptive theorem-prover [1989] (see section 7.4 for more discussion of his procedure).

Fixing of functions is closely related to how circumscription affects equality. [Rathmann, 1989] discusses equality in circumscription. Our primary focus in this thesis is not on that issue.

2.4 Conflict and the Concept of Prioritization

In sections 1.5 and 1.6, we gave examples (1.4, 1.5, and 1.6), at an intuitive level, of conflict and prioritization-type preference between defaults. In this section, we make those concepts more precise, and formulate the Quaker-Republican examples in terms of prioritized predicate circumscription.

Example 2.25 (Non-Monotonicity, In Predicate Circumscription)
Let us formulate Example 1.3 in terms of predicate circumscription. For this purpose, it is convenient to use a by-now common representational "trick" or device: an "abnormality" predicate.[15] If we think of a default axiom as telling one what is "normally" the case, then its violation is equivalent to "abnormality". Rather than maximizing a (default) elementary formula, one may minimize a newly-introduced "placeholder" abnormality predicate instead. In section 3.2, we will show as a theorem that this results in an equivalent NM theory, for the rest of the base language, even when prioritization is considered.

Let the base sentence $B1$ consist of the for-sure sentence that:

$$Republican(Nixon)$$

along with an axiom that expresses the definition of an abnormality predicate $ab1$:

$$\forall x.\, \neg ab1(x) \;\equiv\; (Republican(x) \supset \neg Pacifist(x))$$

Let the minimized pre-order $H1$ be the predicate pre-order corresponding to $ab1$:

$$Z' \preceq_{H1} Z \;\stackrel{\text{def}}{\equiv}\; ab1' \leq ab1$$

Let $\mathcal{NMTH}(\mathcal{A}_0)$ in Example 1.3 be represented as the circumscriptive theory for the circumscription

$$C(B1; H1; Z)$$

[15]See [McCarthy, 1986] and [Lifschitz, 1984] for its historical introduction.

where Z is the tuple of all predicate symbols[16] in the base language (here, $\langle ab1, Republican, Pacifist \rangle$). [17] Then, as desired[18]:

$$\mathcal{A}_0 \;\; \models\!\!\!\approx \;\; \neg Pacifist(Nixon)$$

The update that *Nixon* for-sure is a Pacifist can be represented by adding that fact to the base sentence. Let $B2$ consist of $B1$ conjoined with

$$Pacifist(Nixon)$$

We then represent $\mathcal{NMTH}(\mathcal{A}_0 \& \mathcal{U}_1)$ via[19]

$$C(B2; H1; Z)$$

Then, as desired:

$$\mathcal{NMTH}(\mathcal{A}_0 \& \mathcal{U}_1) \;\; \not\models\!\!\!\approx \;\; \neg Pacifist(Nixon)$$

This illustrates the non-monotonicity of circumscription.

Next, we formulate Example 1.4 in terms of *parallel* predicate circumscription: in which several predicates are minimized "in parallel". "Parallel" can be viewed as the special case of prioritization in which the prioritization is empty. I.e., no strict precedence exists between the minimized predicates / defaults. Historically, parallel circumscription was developed by Lifschitz and McCarthy before they developed the idea of prioritization.

Definition 2.26 (Parallel Predicate Circumscription)
In *parallel* predicate circumscription, the minimized pre-order is simply the conjunction of several predicate pre-orders.[20]

Example 2.27 (Conflict, In Predicate Circumscription)
To represent an axiom set with several defaults, we can introduce an abnormality predicate for each default. Let $B3$ consist of $B1$ conjoined with the axiom that defines a second abnormality predicate $ab2$:

$$\forall x. \neg ab2(x) \;\equiv\; (Quaker(x) \supset Pacifist(x))$$

[16]Remember, we are assuming henceforth that all functions are fixed. Accordingly, as a convention, we will be omitting them from the notation for pre-orders and their arguments.

[17]The notation of [Lifschitz, 1985] is more common in the circumscription literature: there, the above circumscription is written as $C(B1; ab1; Republican, Pacifist)$. In this notation, the second field lists which predicates are minimized (see below for explanation of how there may be several), and the third lists which other predicates are varied (not fixed). In that notation also, all functions are fixed by convention.

[18]the proof of this example, and of the rest of the examples of circumscriptive entailment in this section, are straightforward, using theorems in [Lifschitz, 1985]

[19]Remember, as a convention, we will usually identify the circumscriptive theory with the circumscription, when the distinction between them is not important.

[20]See Definition 2.44 in sub-section 2.7.1 for the more general concept of parallel prioritization.

as well as the for-sure fact

$$Quaker(Nixon)$$

Let $H2$ be the parallel predicate pre-order expressing the minimization of both abnormality predicates "in parallel":

$$Z' \preceq_{H2} Z \stackrel{\text{def}}{\equiv} (ab1' \leq ab1) \wedge (ab2' \leq ab2)$$

We formulate $\mathcal{NMTH}(\mathcal{A}_0 \& \mathcal{U}_2)$ as [21]

$$C(B3; H2; Z)$$

Then, as desired:

$$\mathcal{NMTH}(\mathcal{A}_0 \& \mathcal{U}_2) \not\approx \neg Pacifist(Nixon)$$

$$\mathcal{NMTH}(\mathcal{A}_0 \& \mathcal{U}_2) \not\approx Pacifist(Nixon)$$

The minimizations of the two abnormality predicates, corresponding to the two defaults, conflict in the following sense.

Definition 2.28 (Conflict Between Defaults)
Relative to a given NM axiom set in a NM formalism for default reasoning: We say that two or more defaults *conflict* when, given the for-sure axioms, one cannot consistently believe them all (the for-sure and the defaults, simultaneously) as conclusions.

For more about the concept of conflict, see Theorem 5.75 (in sub-section 5.13.3), and the lemmas used in its proof (in the Appendix).

In our earlier examples (1.5 about pacifism and 1.6 about commuting) and discussion of precedence, we left its definition rather vague. Next, we formulate the precedence-type preference in Example 1.5 in terms of prioritization.

Example 2.29 (Prioritization, Quakers and Republicans)
The preference for the *Republican* default over the *Quaker* default can be represented in *prioritized* predicate circumscription [Lifschitz, 1985] by making the minimization of its corresponding abnormality predicate, $ab1$, have higher priority than the minimization of the other's, $ab2$. This is accomplished by modifying the minimized pre-order $H2$ to be $H3$ instead, where

$$Z' \preceq_{H3} Z \stackrel{\text{def}}{\equiv} \{[ab1' \leq ab1] \wedge [(ab1' = ab1) \supset (ab2' \leq ab2)]\}$$

This is a kind of lexicographic ordering of the two individual predicate pre-orders, where the higher priority criterion comes first in that ordering.

[21] In [Lifschitz, 1985]'s notation: $C(B3; ab1, ab2; Republican, Quaker, Pacifist)$

Observation: This lexicographic composition is the basic intuition behind the concept of prioritization. We represent $\mathcal{NMTH}(\mathcal{A}_0 \& \mathcal{U}_2 \& \mathcal{U}_3)$ as [22]

$$C(B3; H3; Z)$$

Then, as desired:

$$\mathcal{NMTH}(\mathcal{A}_0 \& \mathcal{U}_2 \& \mathcal{U}_3) \;\;\approx\;\; \neg Pacifist(Nixon)$$

Analytic Perspective: Aggregating Model Preference Criteria:
In Examples 2.27 and 2.29, the minimization of each abnormality predicate can be viewed as corresponding to a preference criterion on models. The first default corresponds to a preference for models that make more (in the sense of set inclusion) $Republican$s be non-$Pacifist$s(more precisely, for models that have a smaller extension of the predicate $ab1$, and, equivalently, a larger extension of the wff
$Republican(x) \supset \neg Pacifist(x)$). The second default corresponds to a preference for models that make more $Quaker$s be $Pacifist$s. The prioritization, initially empty, then strict, provides a basis for *aggregating* (i.e., combining) the two preference criteria into a single, overall, "global" preference criterion. Initially, that global preference criterion is indifferent between models that make $Nixon$ a $Pacifist$ and those that make him a non-$Pacifist$. After the strict prioritization is established, though, the latter kind are preferred (strictly) by the global preference criterion. Prioritization resolves the conflict about $Nixon$'s pacifism among the two preference criteria.

In section 2.7, we will see precisely how prioritization is defined for more than two predicates / defaults. More generally, prioritization may resolve some, none, or all of the conflict in a default theory. Indeed, this provides an alternative way that we might think about defining conflict: to say that conflict is present iff adding prioritization alters the NM theory.

2.4.1 Specificity and Inheritance: Review

Note to Reader: If you are already familiar with the concepts of specificity and inheritance, you can skip this sub-section with no loss.

Specificity was the basis for prioritization in the motivating examples that [McCarthy, 1986] and [Lifschitz, 1985] gave when originally developing the idea of prioritization. Specificity arises especially in the context of default inheritance, which is an important pattern of inference. Default inheritance is a central and widely-used mechanism in practical AI Frame-based systems (e.g., recall footnote on page 15) and in many object-oriented programming languages. [Touretzky, 1986] is a good starting point to learn about it.

Example 2.30 (Specificity and Inheritance: Elephants)

[22] In [Lifschitz, 1985]'s notation: $C(B3; ab2 > ab1; Republican, Quaker, Pacifist)$

Consider the following axiom set and normative pattern of inferences. By default, elephants have gray skin. For-sure, albino elephants are a kind of elephant. But, by default, albino elephants have white skin. For-sure, skin cannot both be gray and white. For-sure, Clyde is an elephant. From these axioms, we would like to conclude that Clyde has gray skin. That is, we would like for Clyde to *inherit* his skin-color property from the class of elephants. But we would like this inheritance to be defeasible. After all, we might learn the for-sure information that Clyde has been tattooed and is therefore purple. We would also like this inheritance to be defeasible in another way: suppose we learn that, for-sure, Clyde is an albino elephant. Now, we would like to conclude that Clyde has white skin. This is because we would like him to inherit his skin-color property from the albino-elephant class. Why? The albino-elephant class is more *specific* than the elephant class. It is an overriding exception condition, with respect to elephants and skin-color. Its associated default, therefore, should take precedence over the less specific class' associated default.

The above example can be represented straightforwardly in prioritized predicate circumscription, in a manner very similar to Example 2.29. We will not take the space to go through the entire example, but rather just specify the circumscription for the last axiom set in the above story. Let base sentence B consist of:

$$albino\_elephant(Clyde)$$

$$\forall x.\, albino\_elephant(x) \supset elephant(x)$$

$$\forall x.\, \neg ab1 \ \equiv \ (elephant(x) \supset gray\_skin(x))$$

$$\forall x.\, \neg ab2(x) \ \equiv \ (albino\_elephant(x) \supset white\_skin(x))$$

$$\forall x.\, \neg(gray\_skin(x) \wedge white\_skin(x))$$

Let the minimized pre-order H be the prioritized predicate pre-order corresponding to the minimization of $ab1$ and $ab2$, with the minimization of $ab2$ being at higher priority than $ab1$.

A whole sub-community of the NMR research community has been, and continues to be, occupied with default inheritance. Some cases have problematic intuitions regarding just what should be the normative pattern of inference. The above example illustrates the one case that is, normatively, the most basic and un-problematic, however. In this case, there is a chain of 2 or more "classes" (Ci's) (here albino-elephants and elephants), each related to the ones above by for-sure implication (subsumption). (I.e., $Cj \leq Ci$, for j greater than i.) Each class Ci may have an associated default of the form "if $Ci(x)$, then $Ai(x)$", where the Ai's are logically independent of the Ci's. The Ai and Aj for $i \neq j$ are often mutually exclusive (contradictory) properties. [Etherington and Reiter, 1983] provides an intuitively nice formulation of this kind of case in terms of Default Logic.

2.5 Bases for Prioritization Information

Our intuition is that prioritization is a tool to specify relative confidence or precedence. In considering past and potential applications of prioritization, we have found several kinds of bases of prioritization information.

1. The specificity dominance principle employed in inheritance, e.g., cf. [Touretzky, 1986] [Stein, 1989], conditional logics, e.g., cf. [Delgrande, 1987a] [Delgrande, 1987b] [Geffner, 1992], and argument systems, e.g., cf. [Loui, 1987]. Specificity was the basis for prioritization in the motivating examples that [McCarthy, 1986] and [Lifschitz, 1985] gave when originally developing the idea of prioritization.

2. Reliability of sources. A source might be a communicating agent, or perhaps a sensor. In Example 1.6, reliability was a basis for precedence. Another example is that, say, from a child's point of view, "things my mother told me" have precedence over "things my brother told me", which in turn have precedence over "things the school bully told me". [Lifschitz, 1988a] gives a few examples of reliability as an intuitive basis for prioritization.

3. Recursive depth in programming language semantics: for example, in the semantics of negation as failure in stratified logic programs with negation [Lifschitz, 1987a] [Lifschitz, 1988b] [Przymusinski, 1988].

4. Authority in the legal and organizational senses. For example, federal law takes precedence over state law; policy directives from the head of an organization take precedence over those from subordinates.

5. Temporal precedence. For example, when updating a conventional database, information with a later timestamp typically takes precedence over that with an earlier. This kind of temporal precedence can be viewed as a special case of reliability: fresher reports are regarded as more reliable. In Example 1.6, this was the basis for the precedence of the information in the phone report from Jon's wife over the information in the note. Chronological minimization [Shoham, 1988] is another example of temporal precedence (though, interestingly, in the opposite direction of time as basis for higher precedence!).

6. Difference in (probabilistically) expected utility of actions based on defaults. For example, an action policy oriented toward an emergency situation has precedence over one oriented toward a routine situation, since in the case when both apply (i.e., in an emergency), following the first policy but not the second has higher expected utility than vice versa.

2.6 A Motivating Example

Our generalization of prioritization is motivated by the desire to represent the above kinds of precedence. The previous definition of prioritization, we show, is expressively inadequate for these purposes.

Example 2.31 (Meetings)

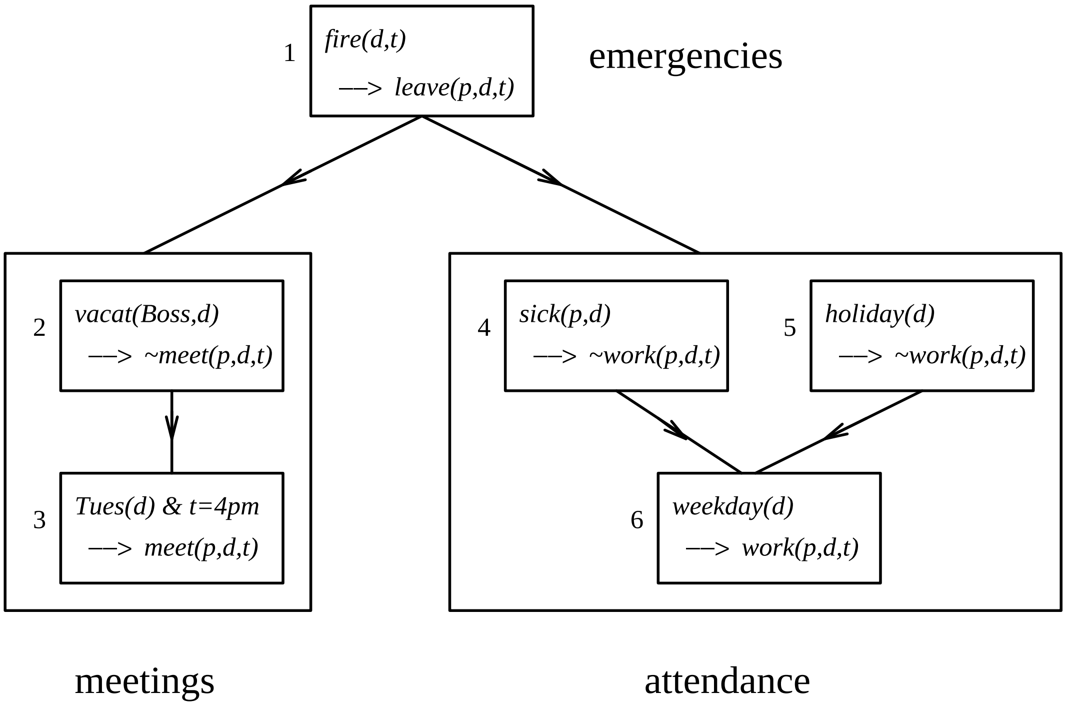


Figure 2.1: "Meetings" example with non-layered precedence. Formulas represent default axioms; the :> prefixes are omitted. Each number represents a default axiom label. p, d, and t vary over persons, days, and times, respectively.

Figure 2.1 illustrates an example of "common-sense" knowledge that a worker in a particular company might have. Some of her default beliefs are about meetings. For example, on Tuesdays at 4pm, normally, there is a meeting of her work unit including her boss; however, if the boss is on vacation, then, normally, there is no meeting. Other of her default beliefs are about work attendance. For example, on weekdays, normally, she goes to work. However, normally, if she is sick, then she does not go to work. Likewise, she normally does not go to work when it is a holiday. In addition, she has default beliefs about emergencies, for example that she leaves the building if there is a fire.

In this example, it is natural to think of the precedence information as being defined partly *within* groups of defaults, and partly *between* those groups. When one group takes precedence over a second, every default in the first takes precedence over every default in the second. Thus the partial order of precedence between groups, *composed* with the partial orders of precedence within each group, leads to a partial order of precedence among individual defaults. More generally, we can imagine that groups may be formed of sub-groups. An individual default can be viewed as a primitive group of one.[23]

In our example, the defaults about meetings form one group. Within that, the vacation rule takes precedence over the Tuesday-4pm rule. The defaults about work attendance form another group. Within that, the sick rule and the holiday rule each take precedence over the weekday rule;

[23] Later, we give a precise, formal definition (2.51) of composing precedence partial orders.

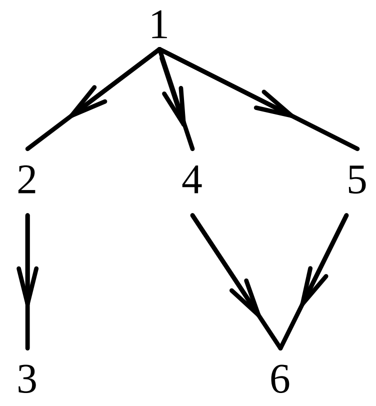


Figure 2.2: Dag view of precedence among the individual defaults in the Meetings example from the last Figure

however, there is no precedence between the sick rule and the holiday rule. Finally, the defaults about emergencies form another group (there might be, say, defaults about what to do in case of earthquakes, violence, heart attacks, etc.; we do not show them to keep this example briefer). The *group* of rules about emergencies takes precedence both over the group of rules about meetings, and over the group of rules about attendance. There is no precedence between the meetings group and the attendance group. Figure 2.1 shows, as a directed acyclic graph (dag), the partial order of precedence between the groups of defaults.[24] We treat an individual default as a primitive group. Groups are enclosed in boxes. A directed arc indicates that the *from* group takes precedence over the *to* group. Figure 2.2 shows, as a dag, the corresponding partial order of precedence among the individual defaults that results from composition.

Observation 2.32 (Shortcomings in Previous Definition)
Now, suppose we wish to represent the default and precedence information in the above example using the tool of prioritization. There turn out to be two problems with the previous definition of prioritization, given in [Lifschitz, 1985]. The first problem is that it is impossible to represent the partial order of precedence in our example (Figure 2.2). Lifschitz' definition only permits partial orders that are finite and what we call *layered*, i.e., that have a structure similar to the system of rank in the military. (Some others have called this "stratified".[25]) Figure 2.3 (a), and the equivalent Figure 2.3 (b), illustrate a layered partial order. Viewed as a dag, a layered partial order consists of a totally-ordered series of one or more levels. At each level, there are one or more elements, with no links between them. [26] Our example (Figure 2.2), however, is *non*-layered.

[24]Recall that any finite strict partial order is isomorphic to a dag.

[25]We chose to employ the new terminology "layered" partly because the term "stratified" has previously been used in some of the non-monotonic reasoning literature (e.g., [Brown and Shoham, 1989]) in the sense of totally-ordered. Also, we wanted to emphasize the prioritization as given, whereas the usage of "stratified" in the logic programming literature (e.g., [Lifschitz, 1987a] [Lifschitz, 1988b] [Przymusinski, 1988]) involves a non-deterministic choice of a particular stratification given a collection of constraints.

[26]See sub-sections 2.7.2 and 2.7.3 for precise definition, and more discussion, of layered-ness.

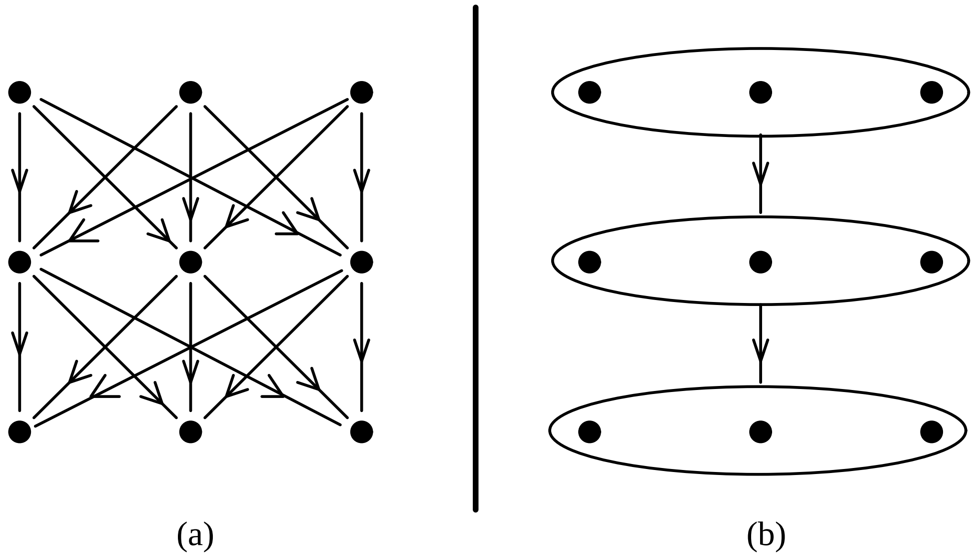


Figure 2.3: A layered p.o. (a) in detail: as dag; (b) via composition: as dag over groups

The precedence among the defaults in the commuting story (Example 1.6) is also non-layered, as illustrated in Figure 1.2.

The second problem is that Lifschitz defined prioritization only over *individual* minimized predicates (in effect, corresponding to individual defaults). His definition does not permit one to define prioritization directly over *groups* of defaults.

These two difficulties with Lifschitz' definition arise often. Specificity and reliability, for example, are often not layered. It is natural to specify precedence among groups that are associated with, for example, sources or topics.

Next, we show how to overcome these difficulties by formally generalizing the idea of prioritization: firstly, to be defined for non-layered partial orders of precedence; and secondly, to be defined (hierarchically) over groups (and groups of groups, etc.) of defaults, i.e., to be composable.

2.7 Generalizing the Definition of Prioritization

Next, we define prioritization as an *operation* on an indexed set of arbitrary pre-orders, that produces a new pre-order. Prioritization, thus, is a tool to define interesting pre-orders. We will use it to define a generalized idea of prioritized circumscription. In particular, we will consider three kinds of "input" pre-orders to the prioritization operation: predicate pre-orders; default (elementary formula) pre-orders; and pre-orders that are themselves the product of prioritization.[27]

However, we view prioritization abstractly: as a method to *aggregate preference criteria*. In our

[27] A Historical Note: Our definition of prioritization was developed in 1986, first presented publicly at the Stanford University Non-Monotonic Reasoning Seminar in spring of 1987, and first published (for the prioritized-default case) in February 1989 in the IBM Research Report version of [Grosof and Russell, 1990], which also included the definition of the Circumscriptive Language of Defaults (discussed in section 3.2).

view, it is not tied to the particulars of predicates and circumscription.

Intuitively, prioritization is a kind of generalization of a lexicographic ordering.

Definition 2.33 (Prioritized Pre-Order)
Let $H \stackrel{\text{def}}{=} \{Hi \mid i \in N\}$ be an indexed tuple of pre-orders, where N is the tuple of *indices*, and where each Hi is defined over the same common domain $\mathcal{D}$. We permit any domain: we do not require that it be the models of a logical language, or the symbol tuples (e.g., Z in Definition 2.7) that (often) correspond to models. Let R be a strict partial order defined over N, where $R(j,i)$ means that index j has *higher priority* than index i. We call R a *prioritization* partial order. (R represents a partial order of precedence.) Furthermore, let R be well-founded (in the direction of ascending priority).[28] Note that any (strict) partial order defined over a finite domain is well-founded; therefore, finiteness of N suffices. We define the *prioritized pre-order* $(H;R)$ as:

$$Z \preceq_{(H;R)} Z' \stackrel{\text{def}}{\equiv}$$

$$\forall i \in N.\ [\forall j \in N.\ R(j,i) \supset (Z \approx_{Hj} Z')] \ \supset\ (Z \preceq_{Hi} Z')$$

The **resulting** prioritized pre-order $(H;R)$ is defined over the same domain $\mathcal{D}$ as the **starting** pre-orders Hi **(terminology)**.

You may get some intuition about this definition from the following analogy. View the index $i \in N$ of each starting pre-order Hi as a voter. $Z \preceq_{Hi} Z'$ means that voter i votes non-strictly (i.e., "agrees") in favor of Z'. Then in order for the overall preference function (i.e., $Z \preceq_{(H;R)} Z'$) to be (non-strictly) in favor of Z', each voter must vote (non-strictly) in favor in case of a tie (i.e., $\approx_{Hj}$) by all the voters who are higher-priority ($R(j,i)$) than that voter. Thus, if a voter's "higher-ups" do not tie, then that voter is "of no account": his agreement is not required.

Theorem 2.34 (Well-Definition)
Prioritization according to Definition 2.33 is well-defined for well-founded prioritization partial orders.

Let R be a well-founded prioritization defined over N.[29] Let H be a tuple of pre-orders, indexed by N, each defined over a common domain. Then $(H;R)$ is a pre-order, defined over the same domain.

Proof Overview: Quite non-trivial. Reflexivity of $(H;R)$ is easy to show. However, our proof of the transitivity of $(H;R)$ is more complicated: it is inductive, using the well-foundedness of R. See Appendix for full proof. □

[28] A partial order S (strict or non-strict) is **defined** to be *well-founded* when it has no infinite descending chains. The usual mathematical convention is that $S(i,j)$ means that i is less than (strictly or non-strictly) j. Here, we are using the opposite interpretation. We require that there be no infinite chains in the direction of ascending priority.

[29] Actually, R could be defined beyond N. More generally and precisely, then, let the restriction of R to N be well-founded.

Observation 2.35 (Finite Prioritizations)

Any finite partial order is well-founded. Therefore, prioritization according to Definition 2.33 is always well-defined for finite prioritization partial orders.

Observation 2.36 (Every Pre-Order Is Prioritized)

Any pre-order is a prioritized pre-order in the trivial or degenerate sense: let the index set N be a singleton. (The prioritization partial order R is then the empty relation.)

The well-foundedness condition is necessary as well as sufficient.

Theorem 2.37 (Ill-Definition)

Prioritization according to Definition 2.33 fails to be well-defined when the prioritization partial order is not well-founded.

For any non-well-founded prioritization, there is a starting set of pre-orders for which the result of prioritization according to Definition 2.33 fails to be transitive.

Proof Overview: See Appendix for the proof by counterexample. □

In [Grosof, 1992b], we discuss how to modify the definition of prioritization to behave well on a large class of non-well-founded prioritization partial orders, including the counterexample given in the proof.

2.7.1 Generalizing Prioritized Circumscription

> I expect that prioritized circumscription will turn out to be the most natural and powerful variant [of circumscription]. *John McCarthy* [1986] (p. 105).

Definition 2.33 was defined for arbitrary pre-orders on arbitrary domains. Next, we apply it, in particular, to generalize the definition of prioritized circumscription.

We define generalized *prioritized circumscription* as, simply, *circumscription with respect to a prioritized pre-order*.

Definition 2.38 (Prioritized Circumscription)

As before, let Z be the tuple of all predicates in $\mathcal{L}$. Let H be a tuple of pre-orders defined over Z, indexed by N. Let R be a prioritization partial order defined over N. Then the *prioritized circumscription* of $B[Z]$ with respect to the prioritized pre-order $(H;R)$, is:

$$C(B;H;R;Z) \quad \stackrel{\text{def}}{\equiv} \quad B[Z] \ \wedge \ \neg\exists Z'.\, B[Z'] \ \wedge \ \ Z' \prec_{(H;R)} Z$$

Observation 2.39 (Every Circumscription Is Prioritized)

Since, as we remarked earlier (Observation 2.36), every pre-order is a prioritized pre-order in a trivial sense: every circumscription, too, is a prioritized circumscription.

N and R are meta-linguistic:
Note that N and R are meta-linguistic relative to the base language $\mathcal{L}$; they are not predicates in $\mathcal{L}$. Elsewhere [Grosof, 1992a], however, we do introduce N are R into the base language, so as to defined reasoning about prioritization and pointwise specification of prioritization, building upon the spirit of [Lifschitz, 1988a].

One interesting kind of starting pre-order, most familiar in previous work on circumscription, is an individual predicate pre-order.

Definition 2.40 (Prioritized Predicate Pre-Order)
We define a (generalized) *prioritized predicate pre-order* $(P^N; R)$, where P^N is a tuple of predicates, indexed by N, as the special case of a prioritized pre-order where each pre-order Hi is an individual predicate pre-order. When there are only a finite number[30] of predicate symbols in $\mathcal{L}$, then N must be finite and, therefore, $(P^N; R)$ is well-defined.

For example,

$$Z \preceq_{P;R} Z' \stackrel{\text{def}}{\equiv} \{P1 \leq P1' \wedge (P1 = P1' \supset P2 \leq P2') \wedge P3 \leq P3' \wedge (P3 = P3' \supset P4 \leq P4')\}$$

is the prioritized predicate pre-order formed by prioritizing the four individual predicate pre-orders $Pi \leq Pi'$, for $i = 1, 2, 3, 4$, according to the prioritization R in which 1 has higher priority than 2 and 3 has higher priority than 4. For example, $P1$ might correspond (i.e., as an "abnormality"; recall section 2.4 and see section 3.3) to the default that ostriches do not fly, $P2$ might correspond to the default that birds do fly, $P4$ to the default that on Tuesdays one must park on the right side of the street, and $P3$ that on holidays one may park on either side of the street.

Definition 2.41 (Prioritized Predicate Circumscription)
We then define a *prioritized predicate circumscription* $PPC(B; P^N; R; Z)$, where $P^N \subseteq Z$, as circumscription with respect to a prioritized predicate pre-order $(P^N; R)$. Note that in this notation, all of Z is varied.[31]

We give an example of a prioritized predicate pre-order and circumscription in section 2.8.

A second interesting kind of starting pre-order is a default, or elementary formula, pre-order.

Definition 2.42 (Prioritized Default Pre-Order)
We define a *prioritized-default pre-order* $(D^N; R)$, where D^N is a tuple of default formulas, as the special case of a prioritized pre-order where each pre-order Hi is an individual default pre-order.

For example, in the Quaker-Republican example with priorities (Example 1.5), we can represent each of the two defaults as a starting default pre-order. The prioritized default pre-order that results

[30] a common restriction in AI knowledge representation when working with finitely-axiomatized theories

[31] See sections 2.3 and 3.5 about fixing.

is:

$$
\begin{aligned}
&Z^1 \preceq_{D;R} Z^2 \quad \stackrel{\text{def}}{\equiv} \\
&\quad [\forall x.\ (Rep^1(x) \supset \neg Pac^1(x)) \ \supset\ (Rep^2(x) \supset \neg Pac^2(x))] \ \ \wedge \\
&\qquad \{[\forall x.\ (Rep^1(x) \supset \neg Pac^1(x)) \ \equiv\ (Rep^2(x) \supset \neg Pac^2(x))] \\
&\qquad\qquad \supset\ \ [\forall x.\ (Qua^1(x) \supset Pac^1(x)) \ \supset\ (Qua^2(x) \supset Pac^2(x))]\}
\end{aligned}
$$

where we have abbreviated the predicate symbols to three letters. (Notational reminder: remember that Z^k, for $k = 1, 2$, is defined as
$\langle Qua^k, Rep^k, Pac^k \rangle$.)
Intuitively, this prioritized default pre-order says that Z^2 is preferred (non-strictly, with respect to the default pre-order) to Z^1 iff: Z^2 makes at least as many Republicans be non-Pacifists as does Z^1, and, when there is a tie between Z^2 and Z^1 in that regard, Z^2 makes at least as many Quakers be Pacifists as does Z^1. (Here, by "at least as many", we mean in the sense of superset).

Note that, since a predicate pre-order is a special case of a default pre-order, a **prioritized predicate pre-order is a special case** of a prioritized default pre-order.

Definition 2.43 (Prioritized Default Circumscription)
We define a **Prioritized Formula Circumscription** $PFC(B; E; R; Z)$ as circumscription with respect to a prioritized default pre-order $(E^N; R)$. "Formula" here means elementary formula. Remember that since circumscription is defined in terms of minimization, the maximized defaults here are the negations of the Ei's. What McCarthy [1986] defined as "formula circumscription" is the special case where there is only one elementary formula pre-order. The reason we speak of "elementary" formulas is that a prioritized elementary-formula pre-order is itself a formula; we wish to avoid confusion.

Usually in this thesis, we will find it a bit more convenient to speak in terms of a notation for prioritized default circumscription in which the default formulas appear as **maximized instead** of minimized. We, accordingly, define the **Prioritized Default Circumscription (PDC)** $PDC(B; D; R; Z)$ as circumscription with respect to the prioritized default pre-order $((\neg D)^N; R)$, i.e., as $PFC(B; \neg D; R; Z)$. Note that in both the PFC and the PDC notation, each Ei or Di is presumed to be defined in terms of symbols Z, and that all of Z is varied.[32] We will discuss the representational use of prioritized default circumscription starting in chapter 3.

More explicitly:

$$PDC(B; D; R; Z) \ \equiv\ B[Z] \ \wedge\ \neg\exists Z'.\ B[Z'] \wedge Z \prec_{(D;R)} Z'$$

where $\prec_{(D;R)}$ is defined as the strict version $(\preceq_{(D;R)} \wedge \neg\succeq_{(D;R)})$ of the prioritized "formula" pre-order $\preceq_{(D;R)}$:

$$
\begin{aligned}
&Z \preceq_{(D;R)} Z' \quad \stackrel{\text{def}}{\equiv} \\
&\quad \forall i \in N.\ [\forall j \in N.\ R(j,i) \supset (\forall xi.\ Dj[Z, xj] \equiv Dj[Z', xj])] \\
&\qquad\qquad \supset\ (\forall x.\ Di[Z, xi] \supset Di[Z', xi])
\end{aligned}
$$

[32] See sub-section 2.3.1 and section 3.5 about fixing, though.

Here Dj and Di refer to the j^{th} and i^{th} members, respectively, of the tuple D. xj and xi are each a tuple, of individual object variables, of the appropriate arity for Dj and Di, respectively.

We **define** the corresponding **prioritized default theory** to be the circumscriptive theory corresponding to the prioritized default circumscription.

Note that **prioritized predicate circumscription is the special case** of prioritized default circumscription where each maximized default formula is a negated open literal of the form $\neg P(x)$, i.e., where the defaults are the negations of predicates.

An important special case of prioritization is when there are no strict precedence relationships at all: the "parallel" case. In this case, the prioritized pre-order is simply the conjunction of the starting pre-orders.

Definition 2.44 (Parallel Priority)
When a prioritization partial order relation R is false everywhere, we say that it is *empty* and write it as $\emptyset$. We also call this *parallel* prioritization.

A parallel predicate or default circumscription is one with parallel priority.

Observation 2.45 (Finite N)
Recall our assumption (page 27) that the base language, and our axiomatizations (e.g., the axiom sets expressed in the Circumscriptive Language of Defaults, defined in chapter 3) are finite. Thus N is too: in prioritized predicate and prioritized default circumscriptions, unless explicitly stated otherwise.
$(P^N; R)$ and $(E^N; R)$ $((D^N; R))$,

Definition 2.46 (Instantiation of Defaults)
Let $D1[Z, x]$ be an open default formula, where $x \stackrel{\mathrm{def}}{=} \langle x1, \ldots, xm \rangle$ is a tuple of individual (object) variables, of length m. Let $t \stackrel{\mathrm{def}}{=} \langle t1, \ldots, tm \rangle$ be a tuple of ground terms, also of length m. Then we say that $D1[Z, t]$ is a *default instance* or, simply, an *instantiated default*.

More generally, the terms ti need not be ground. Indeed, each xi qualifies as a term. Then we say that $D1[Z, t]$ is a *partially instantiated* default formula. Note that $D1[Z, x]$ is a partially instantiated version of itself, though of course in a degenerate sense. When t consists only of *ground* terms, then we say that the instantiation is *full* or *complete*. Full instantiation is thus a special case of partial instantiation.

Theorem 2.47 (Open Default As Parallel Default Instances)
In any prioritized default circumscription (PDC):
Suppose that the base sentence entails domain closure. Suppose also that all functions are fixed or, more generally, fixed "relative" to the circumscription (see Definition 3.47). Then every open default is equivalent to the parallel prioritization of the tuple of all its default instances. I.e., in the context of the circumscription, every (open default's) default pre-order is equivalent to the conjunction of the default pre-orders for that default's instances.

More explicitly: for each $i \in N$,

$$\forall xi.\ Di[Z, xi] \supset Di[Z', xi] \ \equiv$$
$$(Di[Z, t1] \supset Di[Z', t1]) \ \wedge \ \ldots \ \wedge \ (Di[Z, tm] \supset Di[Z', tm])$$

where $t1, \ldots, tm$ are all of the tuples, of the domain elements, of the appropriate arity for Di.

Note that uniqueness of names, in the presence of domain closure, suffices to effectively fix (i.e., fix "relative") all the functions, by Theorem 2.24.

Note also that the sufficient condition above can be weakened.

See Lemma 3.23 for the generalization of this theorem to the "quasi-propositional" case.

Proof Overview: Not terribly complicated. See Appendix. □

2.7.2 The Previous Definition of Prioritization; Layering

Note that we allow *any* finite strict partial order as prioritization partial order. Our definition of prioritized circumscription thus generalizes the previous definition, given by [Lifschitz, 1985]. "Prioritized circumscription" as defined there is the special case of prioritized *predicate* circumscription, where the prioritization partial order R is restricted to be finite and *layered* (what some others have called "stratified"[33]).

In the layered case of prioritized predicate circumscription, the overall set of minimized predicates forms a partition into a totally-ordered set of layers ("strata"). Each predicate in the first layer (stratum) is minimized at higher priority than those in the second layer, which in turn are minimized at higher priority than those in the third, and so on.

The system of military rank is a prototypical example of a layered partial order. E.g., generals have precedence over colonels, who have precedence over majors, etc.. See Figure 2.3 for an illustration of a layered partial order.

Definition 2.48 (Layered Priority)
Let $\{L1, \ldots, Lk\}$ be a partition of N, where $k \geq 1$, such that for each Li, R^{Li} is $\emptyset$, and for each $a \in Li, b \in Lj$. $R(a,b) \Leftrightarrow (i < j)$. Here R^{Li} stands for the restriction of R to the sub-domain Li, and $<$ stands for ordinary arithmetic less-than. Then we say that R is a finitely *layered* (prioritization) partial order, and that the Li's are layers. [34]

We will see later (in chapters 3-7) that the layered case of prioritized circumscription is often simpler than the general in terms of decomposition and other properties. Thus we take some space, next, to give some insight into the expressive characteristics of layering.

One way to view the layered-ness condition is that it places strong constraints on what kinds of *incomparability* are permitted in a (prioritization) partial order.

[33] see footnote on page 45

[34] In [Grosof, 1992b], we define the infinite case of layering.

Definition 2.49 (Incomparable)
We say that two elements i and j of a strict partial order R, are *incomparable* with respect to each other, when neither is greater than the other:

$$\neg R(i,j) \wedge \neg R(j,i) \qquad ;i.e.,$$

$$\neg(i \succ_R j) \wedge \neg(i \prec_R j)$$

We apply the same terminology to pre-orders (including as a special case, non-strict partial orders): when

$$\neg H(i,j) \wedge \neg H(j,i) \qquad ;i.e.,$$

$$\neg(i \preceq_H j) \ \wedge \ \neg(i \succeq_H j)$$

where H is a pre-order.

Observation 2.50 (Properties of Layered)
The following three properties are obeyed by layered prioritization partial orders.

1. Incomparable implies equivalent with respect to priority:

$$\forall a, b \in N.\ \neg R(a,b) \wedge \neg R(b,a) \Rightarrow$$

$$\forall c.\ (R(c,a) \Leftrightarrow R(c,b)) \wedge (R(a,c) \Leftrightarrow R(b,c))$$

2. Incomparable with two implies those two are incomparable:

$$\forall a, b, c \in N.\ \neg R(a,b) \wedge \neg R(b,a) \ \wedge \ \neg R(a,c) \wedge \neg R(c,a) \ \Rightarrow$$

$$\neg R(b,c) \wedge \neg R(c,b)$$

3. i.e.: if one considers the prioritization among the indices $R_I(i)$ which are incomparable to a given index i, then their relative prioritization is empty:

$$\forall i \in N.\ R^{R_I(i)} = \emptyset$$

In terms of military rank precedence: the first property tells us that if two officers cannot pull rank on each other, then they have the same rank relative to everybody who pulls rank on them or that they can pull rank on. The second property tells us that if Joe cannot pull rank on Fred and Barbara, and vice versa, then Fred and Barbara cannot pull rank on each other.

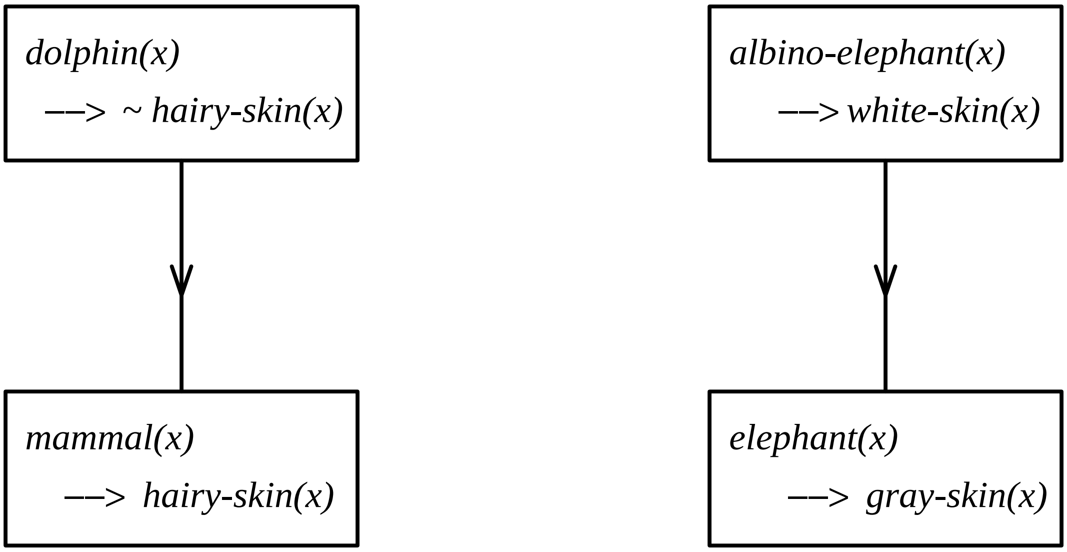


Figure 2.4: An inheritance example with "columnar" prioritization. For-sure, all dolphins are mammals. For-sure, all albino-elephants are elephants. Formulas in the figure stand for default beliefs; the :> and default axiom label prefixes are omitted.

2.7.3 Composing Prioritization

We show next that our definition permits one to *compose* prioritization: the starting pre-orders may themselves be the result of prioritization operations. Thus we can build up a class of prioritized pre-orders *recursively* or *hierarchically*. In particular, we can accomplish the goal we discussed in section 2.6: we can represent a *group* of defaults as one prioritized predicate pre-order, and then represent groups of groups, etc., by composing prioritization. In other words, "a prioritization of prioritizations is always equivalent to a prioritization". Definition 2.51 and Theorem 2.56 below lay out the formal details.

The well-founded class of partial orders is closed, we show, under (a finite number of) composition operations. By contrast, we **observe** that the (finite) layered class is not closed under composition, as we saw in Figures 2.1 and 2.2. The layered class is not closed under composition even when the "composer" partial order is restricted to be parallel, as in Figure 2.4.

Definition 2.51 (Composition of Strict Partial Orders)
Let $R1$ be a strict partial order defined over a domain $N1$ of indices. For each index i in $N1$, let RTi be a strict partial order defined over a domain NTi of indices. Let $N1$ and all of these NTi's be disjoint. Let RT stand for the tuple of the RTi's, and NT stand for the tuple of the NTi's. Then we define the result R of *composing* $R1$ with RT, written $R1{\circ}RT$, to be:

$$\forall a, b \in N.\ R(a,b) \ \stackrel{\text{def}}{\equiv}$$

$$\{ \ \ [\exists i \in N1.\ a \in NTi \ \wedge \ b \in NTi \ \wedge \ RTi(a,b)]$$

$$\vee\ [\exists i, j \in N1.\ a \in NTi\ \wedge\ b \in NTj\ \wedge\ R1(i,j)]\}$$

where

$$N \stackrel{\text{def}}{=} \bigcup_{i \in N1} NTi$$

R is indeed a strict partial order. We say that $R1$ is the *composer* partial order, and that RT is the tuple of *starting* partial orders.

Theorem 2.52 (Closure of Well-Founded Class)
The well-founded class of strict partial orders is closed under composition. (For 1 up to any finite number of composition steps.)

Proof Overview: See Appendix. □

Definition 2.53 (Total Order)
A strict partial order R is *total* when every pair of indices in its domain are comparable:

$$\forall i, j.\ R(i,j) \vee R(j,i)$$

A prototypical example of a total order is arithmetic less-than ($<$) defined over the domain of the natural numbers. This is well-founded. $<$ on the natural numbers $\{1, \ldots, 100\}$ is an example of a finite total order. $<$ on the integers is a non-well-founded total order.

Note that for finite total precedence orders, prioritization is fairly close to the usual idea of a lexicographic ordering.

Definition 2.54 (Columnar Partial Order)
We say a strict partial order is *columnar* when it is equivalent to an empty ("parallel") strict partial order composed with a tuple of total orders. Figure 2.4 illustrates. Columnar precedence among defaults is a common pattern, especially in practical inheritance: see discussion in section 2.9. We also saw, earlier, a columnar structure of precedence in our commuting story.[35]

Observation 2.55 (Layered As Total Composed with Parallel)
Note that any layered partial order is equivalent to the result of composing a *total* order with a tuple of empty (i.e., "parallel") starting partial orders. Figure 2.3 (b) illustrates.

Notation:
It is convenient when discussing composition of prioritization to adopt an in-fix notation for the prioritization operation. We write $R1{*}HT$ to stand for the result of prioritizing a tuple HT of pre-orders, according to a strict prioritization partial order $R1$. (We leave implicit the index tuple

[35] Example 1.6 in sub-section 1.6.2 and Figure 1.2 in sub-section 1.6.3

$N1$ of $R1$). I.e., we let $R1{*}HT$ stand for $(HT; R1)$. We also apply the $\circ$ and $*$ notation to tuples. Thus we write

$$RT \stackrel{\text{def}}{=} RT1{\circ}RTT \qquad ;\, and$$

$$HT \stackrel{\text{def}}{=} RT{*}HTT$$

where RT and $RT1$ are each a tuple of strict partial orders, RTT is a tuple of tuples of strict partial orders, and HTT is a tuple of tuples of pre-orders.

Theorem 2.56 (Composability of Prioritizations)
Prioritization is composable in the following senses. Firstly, prioritizing pre-orders which are themselves the result of prioritization, is equivalent to prioritizing using the composition of the prioritization partial orders involved. One can view this as a kind of associativity property. Formally, using the notation from the above paragraph:

$$R1{*}(RT{*}HTT) \;\equiv\; (R1{\circ}RT){*}HTT$$

(On the right-hand-side, we treat HTT as one big tuple.) Secondly, the result of composing prioritization is well-defined. Formally: Given well-foundedness of the prioritization partial orders ($R1$ and each of the RTi's), the result of composing prioritization is a prioritized pre-order with a well-founded prioritization partial order. By Theorem 2.34, that result is, therefore, a pre-order. This well-definition of composing prioritization is maintained for 1 up to any finite number of composition steps.

Proof Overview: Uses Theorem 2.52. See Appendix. □

2.8 Application to Example: Meetings

Next, we show how to use our generalization of prioritization to represent within prioritized predicate circumscription the Meetings example discussed in section 2.6 and illustrated in Figure 2.1.[36]

Example 2.57 (Meetings, in Prioritized Circumscription)
As in the *Quaker-Republican* formalization (Example 2.29) in section 2.4, let the "base sentence" $B0$ comprise the explicit definition of an "abnormality" predicate corresponding to each of the default formulas:

$\forall p, d, t.\, \neg ab1(p,d,t) \;\equiv\; (fire(d,t) \supset leave(p,d,t))$
$\forall p, d, t.\, \neg ab2(p,d,t) \;\equiv\; (vacat(Boss,d) \supset \neg meet(p,d,t))$
$\forall p, d, t.\, \neg ab3(p,d,t) \;\equiv\; (Tues(d) \wedge t = 4pm \supset meet(p,d,t))$
$\forall p, d, t.\, \neg ab4(p,d,t) \;\equiv\; (sick(p,d) \supset \neg work(p,d,t))$
$\forall p, d, t.\, \neg ab5(p,d,t) \;\equiv\; (holiday(d) \supset \neg work(p,d,t))$

[36]See Example 3.10 in sub-section 3.3.1 for a version of the Meetings example as a prioritized *default* circumscription, expressed in terms of the Circumscriptive Language of Defaults.

$\forall p, d, t.\ \neg ab6(p,d,t) \ \equiv\ (weekday(d) \supset work(p,d,t))$
where the variables p, d, t, should be thought of as varying over persons, days, and times, respectively. Minimizing these abnormality predicates then expresses the maximizing of the corresponding default formulas. We adopt as prioritization R the partial order illustrated in Figure 2.2. The resulting prioritized predicate pre-order $(ab; R)$ is (we omit the $\lambda Z, Z'$. prefix, where the tuple of all symbols Z is $\langle ab1, \ldots, ab6, fire, leave, \ldots \rangle$):
$[ab1' \leq ab1]\ \wedge$
$[(ab1'{=}ab1) \supset (ab2' \leq ab2)]\ \wedge$
$[((ab1'{=}ab1) \wedge (ab2'{=}ab2)) \supset (ab3' \leq ab3)]\ \wedge$
$[(ab1'{=}ab1) \supset (ab4' \leq ab4)]\ \wedge$
$[(ab1'{=}ab1) \supset (ab5' \leq ab5)]\ \wedge$
$[((ab1'{=}ab1) \wedge (ab4'{=}ab4) \wedge (ab5'{=}ab5)) \supset (ab6' \leq ab6)]$

Representational Adequacy:
The overall state of affairs can then be represented via the prioritized predicate circumscription

$$PPC(B0; ab; R; Z)$$

Representational Effect of Non-Layered Prioritization:
Suppose we continue the example with the for-sure information that a person must be at work to be able to meet, that leaving the building prevents a person from working the rest of the day, that today is Tuesday, and thus a weekday. To represent this, let the base sentence $B1$ be the conjunction of $B0$ with the axioms:
$\forall p, d, t.\ meet(p,d,t) \supset work(p,d,t)$
$\forall p, d, t1, t2.\ (leave(p,d,t1) \wedge (t1 \leq t2)) \supset \neg work(p,d,t2)$
$\forall d.\ Tues(d) \supset weekday(d)$
$Tues(Today)$
where $(t1 \leq t2)$ means time $t1$ precedes time $t2$.

Then we can ask questions about what can be concluded by default about employee Ed. E.g., does he have a meeting today at 4pm? The answer turns out to be yes:

$$PPC(B1; ab; R; Z) \models meet(Ed, Today, 4pm)$$

Suppose next we learn that the boss is on vacation today; let $B2$ be the conjunction of $B1$ with:

$$vacat(Boss, Today)$$

Then the default conclusion is that Ed has no meeting after all; the higher-priority default dominates:

$$PPC(B2; ab; R; Z) \models \neg meet(Ed, Today, 4pm)$$

Next, consider the scenario in which Ed is sick with a cold, instead of the boss being on vacation; let $B3$ be the conjunction of $B1$ with:

$$sick(Ed, today)$$

In this situation, the Tuesday-4pm default is in conflict with the sick rule, yet there is no strict prioritization specified one way or the other. Maybe the boss is a slavedriver who expects employees for the weekly meeting unless the illness is deathly. Maybe the company policy puts health first. Who knows? The point is we don't. Accordingly, the prioritized theory leaves the question open and entails no conclusion about whether or not Ed has the meeting:

$$PPC(B3; ab; R; Z) \not\models meet(Ed, Today, 4pm)$$
$$PPC(B3; ab; R; Z) \not\models \neg meet(Ed, Today, 4pm)$$

By contrast, if the prioritization was required to be layered, then we would be forced to conclude definitely one way or the other about Ed having the meeting.

Proof Overview: For proof of this extended example, see Appendix. We give an argument directly in terms of models. □

Composition:

Finally, we remark that the overall prioritized (predicate) pre-order $(ab; R)$ can be viewed as the result of composition. The three groups of defaults, about 1) emergencies (a rather trivial group containing only a single default); 2) meetings (with two defaults); and 3) work attendance (with three defaults) correspond, respectively, to the prioritized (predicate) pre-orders

$$HH1 \stackrel{\text{def}}{\equiv} [ab1' \leq ab1]$$
$$HH2 \stackrel{\text{def}}{\equiv}$$
$$\{[ab2' \leq ab2] \wedge [(ab2' {=} ab2) \supset (ab3' \leq ab3)]\}$$
$$HH3 \stackrel{\text{def}}{\equiv}$$
$$\{[ab4' \leq ab4] \wedge [ab5' \leq ab5] \wedge$$
$$[((ab4' {=} ab4) \wedge (ab5' {=} ab5)) \supset (ab6' \leq ab6)]\}$$

(where, again, we omitted the $\lambda Z, Z'$. prefixes on the right hand sides.)

Let RR stand for the prioritization partial order among these three groups: i.e., that the emergency group (group 1) has higher priority than the meetings group (group 2) and higher priority than the work attendance group (group 3). Then

$$(ab; R) \equiv (HH; RR)$$

where HH is the tuple of the HHi's. In other words, the overall prioritized pre-order can be viewed as the result of composing the prioritization between the groups (RR) with the prioritizations within the groups.

2.9 Why Go Beyond Layered?

2.9.1 Representational Accuracy

One may wonder: "Do I really need to make my life more complicated — can't I get away with always just sticking to the layered case of priority?" Of course, in some particular applications, the appropriate precedence information is layered. More generally, one might try to approximate a non-layered prioritization with a layered one. However, in general, in the context of circumscription, the resulting theory is different: adding a pair (an arc or path in the dag view) to the prioritization relation may result in more conclusions (as more conflict is resolved), while deleting a pair may result in loss of some conclusions. Indeed, for any finite group of prioritized defaults that are logically independent, it is possible to find a base theory B (i.e., a set of for-sure beliefs) that discriminates between any two different given prioritizations. The next two theorems make these observations more precisely.

Theorem 2.58 (Adding Prioritization is Monotonic)
Increasing the prioritization of the minimized pre-order is globally monotonic for any circumscription.

$$[\forall xy.\ R1(x,y) \supset R2(x,y)] \quad \Rightarrow$$
$$[C(B;(H;R2);Z) \ \models\ C(B;(H;R1);Z)]$$

where $R2$ is defined over the same indices (e.g., N) as $R1$.

By "increasing" the prioritization, we mean adding more pairs (paths) to the prioritization partial order (dag).

In the Circumscriptive Language of Defaults, defined in section 3.2, prioritization information is specified via axioms, and accumulates monotone-increasingly. Thus the above implies that:
In CLD, updating with prioritization axioms is globally monotonic.
(Moreover, in terms of Definition 4.17, this monotonicity is "forever".)

Proof Overview: Uses a positivity / negativity argument. See Appendix. □

Theorem 2.59 (Discrimination)
Let D be a finite tuple of default formulas, indexed by N, in symbols Z, that are satisfiable and logically independent of each other. Let $R1$ and $R2$ be two distinct (different) prioritization partial orders defined over N. Then there exists a satisfiable base sentence $B[Z]$ that results in non-equivalent default theories for the two prioritizations; i.e., such that

$$PDC(B;D;R1;Z) \ \not\Leftrightarrow\ PDC(B;D;R2;Z)$$

Proof Overview: Constructive. See Appendix. □

Our basic philosophy is that the aim in knowledge representation is to represent what we know — no less, but also no more. Our knowledge may be quite incomplete, and that includes knowledge about precedence. We want the flexibility to represent a given state of precedence information, which may not be layered, and the open-ness to be able to specify later additional precedence information (which might make it become layered). For example, in Figure 2.1, we would like to be able to leave open the relative precedence of the sick rule and the Tuesday-4pm rule; as we discussed in section 2.8, they might conflict given the for-sure belief that attending work is necessary to be able to meet. If we later learn that the sick rule has precedence, we can represent that by modifying the prioritization: say, to become $R2$ which includes the extra arc. In our final Meetings example in section 2.8, then, we will be able to conclude that Ed does not have to meet today:

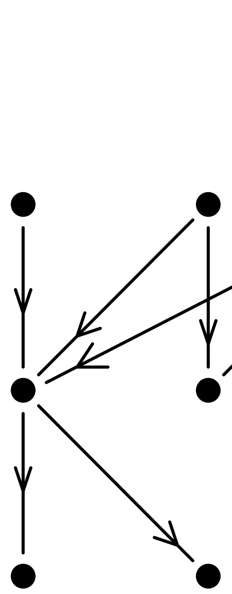


Figure 2.5: A "random" dag: not layered, not columnar, not a lattice.

$$PPC(B3; ab; R2; Z) \models \neg meet(Ed, Today, 4pm)$$

2.9.2 Inheritance and Specificity

Even in relatively simple default inheritance of the kind found in AI "frame-based" systems [Minksy, 1975][37], the appropriate prioritization is non-layered. Common there is a "columnar" structure: several inheritance chains, each about a different topic, with empty (i.e., parallel) prioritization between the chains.[38] Figure 2.4 in subsection 2.7.2 illustrates such a case. Prioritizing in parallel two (or more) groups, each of which internally has total prioritization[39], results in a non-layered overall prioritization.

More generally, in our experience so far in representing specificity dominance in inheritance, we have found that the appropriate prioritization mirrors the directed acyclic graph (dag) of the defeasible links in the inheritance network. Such dags are usually *not* layered. (Recall that any finite strict partial order is isomorphic to a dag.) Figure 2.5 illustrates a "random" non-columnar, non-layered dag of the kind that arises in inheritance networks (e.g., cf. [Touretzky, 1986]). See sub-section 2.11.1 for discussion of how specificity in inheritance can be viewed, in a precise sense, as an implicit specification of prioritization.

Embedding Inheritance in More General Default Reasoning:
Accurately and appropriately representing the epistemologically-justified prioritization is especially important when one tries to embed and to integrate inheritance and specificity into a more general

[37] Recall footnote on page 15.

[38] See Definition 2.54 of "columnar" in sub-section 2.7.3.

[39] By "total", we mean that the prioritization partial order is a total order.

context of default reasoning. When one tries to do more expressive default reasoning that includes inheritance aspects, then "getting the prioritization right" can become a bit subtle. It is easy to find such examples in which no layered prioritization is equivalent to the appropriate, non-layered prioritization. Interestingly, this can arise even when there is only a single inheritance chain! This is because there are in general many instantiations of each of the defaults involved: one for each individual object (e.g., $Peter$ and Tom below). The following example illustrates.[40]

Example 2.60 (Embedding Inheritance)
Consider a tiny inheritance hierarchy, in which we are told that: all students (university students) are adults; typically, adults work; but, typically, students do not work. The specificity dominance principle implies that we would like the student default to take precedence over the adult default. We can represent this in prioritized predicate circumscription by minimizing abnormalities. Let the base $B0$ consist of:

$$\forall x.\ student(x) \supset adult(x)$$

$$\forall x.\ \neg ab1(x) \equiv (adult(x) \supset work(x))$$

$$\forall x.\ \neg ab2(x) \equiv (student(x) \supset \neg work(x))$$

The usual style of previous work on prioritized circumscription, e.g., in [Lifschitz, 1985] has simply made $ab2$ be minimized at higher priority than $ab1$. This is a layered prioritization. However, this does not capture some subtleties of specificity dominance.

Suppose that we are also told the "situation-particular" information that

$$adult(Peter)\ \wedge\ student(Tom)\ \wedge\ (Tom \neq Peter)$$

$$work(Peter) \equiv work(Tom) \tag{EQV1}$$

Strictly speaking, this last axiom (EQV1), saying that either $Peter$ and Tom both $work$ or neither do, cannot be represented in the usual inheritance networks (cf. [Touretzky, 1986]), since it would create a cycle when represented via a pair of ("strict", i.e., for-sure) links. However, such equivalences might easily arise when we *embed* the defaults, corresponding to an inheritance network, within an expressively richer system for more general default reasoning.

With the above layered prioritization, the resulting circumscriptive theory (when we conjoin these last axioms with $B0$ to form the base $B1$) entails that both Tom and $Peter$ do not $work$. However, we[41] find this behavior to be undesirable: we would like the theory to entail nothing one way or the other about whether Tom and $Peter$ $work$. Why? The $student$ default for Tom conflicts with the $adult$ default for $Peter$, since their conclusions are contradictory given (EQV1).

[40] We selected this example because it is about the simplest possible to demonstrate the point. The example was posed (not solved) by Hector Geffner (private communication).

[41] and Geffner

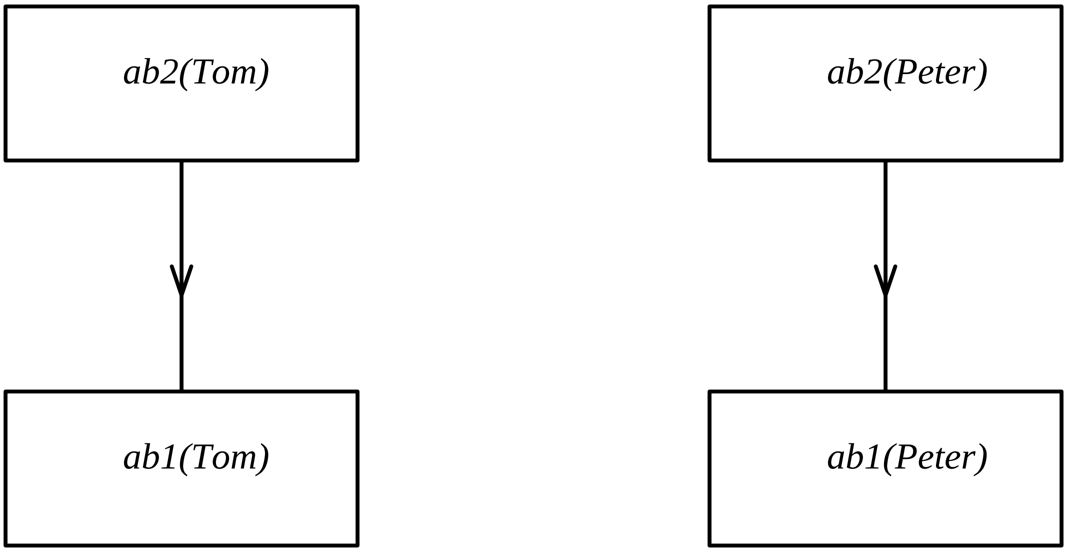


Figure 2.6: Desired, appropriate prioritization for example of embedding inheritance: in terms of (minimized) abnormality-predicate instances.

Intuitively, there is no basis for resolving this conflict: the *student* default for Tom is more specific than the *adult* default for Tom, but it is *not* more specific than the *adult* default for $Peter$. The appropriate prioritization mirrors this specificity structure. Yet the layered prioritization makes the *student* default for Tom have higer priority than not just the *adult* default for Tom, but *also* the *adult* default for $Peter$. This is what resolves the conflict: in our view, contrivedly.

In short, to express the desired theory, we need $ab2(Tom)$ to be minimized at higher priority than $ab1(Tom)$, and $ab2(Peter)$ at higher priority than $ab1(Peter)$, but for the prioritization otherwise to be empty. This requires the prioritization to be non-layered: we can express this straightforwardly via prioritized predicate (or default) circumscription, and get the desired (non-)entailment. Figures 2.6 and 2.7 illustrate. The prioritization happens to be isomorphic to that in Figure 2.4.

Proof Overview: For proof of this example, see Appendix. We give an argument directly in terms of models. □

Observation 2.61 (Prioritization For Embedding Inheritance)
This example illustrates a more abstract principle, not recognized previously in the NMR literature, that applies when embedding inheritance in a more expressive default reasoning scenario. The appropriate prioritization to represent even a single default inheritance chain then involves a parallel composition of many instantiating "replicas" of that chain.

In terms of the example: More generally, considering the universe of individuals beyond just Tom and $Peter$, we would like $ab2(x)$ to be minimized at higher priority than $ab1(x)$ for the *same* x, but neither at higher nor at lower priority than $ab1(y)$ for y different from x. This is not just a non-layered prioritization: it also requires *"pointwise"* expressiveness to specify. In [Grosof, 1992a],

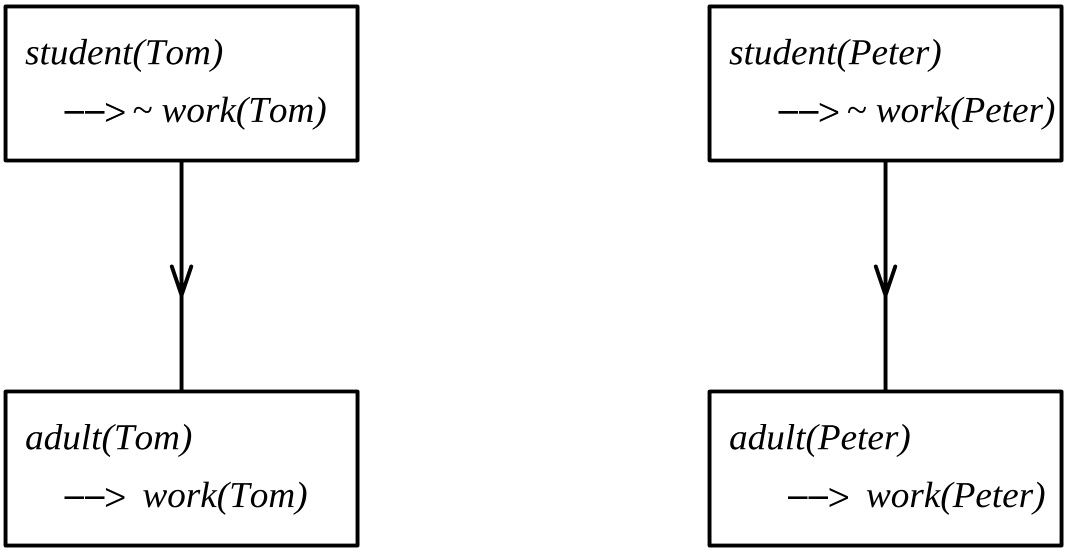


Figure 2.7: Desired, appropriate prioritization for example of embedding inheritance: in terms of (maximized) default-formula instances.

we discuss how to axiomatize prioritization with such pointwise control.

2.9.3 Modules

In representing general default reasoning beyond the field of inheritance, we find that a common pattern is the presence of several "components" corresponding to subsets of our overall set of default beliefs. Often, a component contains beliefs about a *common topic*: more formally, about a syntactically related set of predicates or formulas. While within each component, there is usually some non-empty "internal" prioritization, frequently, there is a common "external" prioritization: i.e., all of the defaults within a component have the same prioritization relative to the defaults outside of the component.

Definition 2.62 (Module)
Thus we can think of these components as being *modules*, where we can *compose modules hierarchically from sub-modules via (composing) prioritization*. A *primitive* module is a single default, which corresponds to a single default[42] pre-order. A module more generally is a prioritized group of defaults, which corresponds to a prioritized default pre-order that may be formed hierarchically or recursively by prioritizing a set of starting pre-orders that themselves represent prioritized (sub-)modules. Thus a module pre-order H is the prioritization of some tuple SH of sub-module pre-orders: $(SH; R)$. In terms of CLD, a module is a collection of default axioms and associated prioritization axioms.

[42]or abnormality predicate

We call a global set of defaults (i.e., the set of all defaults in a global axiom set) plus their associated prioritization: a *global* module. Note that every module pre-order/circumscription $H \stackrel{\text{def}}{=} (SH; R)$ is a prioritized default pre-order/circumscription.

A special case of a module is prioritized group of one or more minimized predicates.

Examples:
The meetings example (Figure 2.1 in section 2.6), the commuting example (Figure 1.2 in sub-section 1.6.3), and the inheritance example (Figure 2.4 in sub-section 2.7.2), displayed modules formed by composing prioritization. There, the defaults were grouped by topic, as we described above.

Corollary 2.63 (Composition of Modules)
It is a corollary of Theorem 2.56 that composition of modules is well-defined, and is equivalent to composition of prioritization. That is: Modules' prioritization composes in the appropriate manner.

More precisely:
Suppose that the global module is the result of composition of modules. Then the composition of modules (in terms of modules being prioritizations of sub-modules) to form the global module can be viewed equivalently as the composition of the modules' prioritization partial orders to form the global prioritization partial order.

Yet more precisely, for the case of one step of composition:

Suppose that the global module pre-order H is the prioritized default pre-order $H \stackrel{\text{def}}{=} (DG^{NG}; RG)$, where

Suppose that the global module is a large-grain-size module that is the (external) prioritization of some mid-grain-size modules:

$$H \stackrel{\text{def}}{=} (RE{*}MOD) \; ;$$

(recall notation in Theorem 2.56) where each mid-grain-size module pre-order is itself the prioritization of some primitive (i.e., "fine-grain-size") modules (i.e., individual default pre-orders):

$$MODj = (DMj^{NMj}; RMj)$$

Then the composition of modules to form the global module can be viewed equivalently as the composition of the modules' prioritization (partial orders) to form the global prioritization (partial order), i.e., in terms of modules being prioritizations of sub-modules:

$$H = (RE{*}(RMT{*}DMT)) = ((RE{\circ}RMT){*}DG)$$

Here, DMT is the tuple of the DMj's. DG stands for DMT (which is a tuple of tuples) treated as one big tuple. DG is thus the global tuple (set) of defaults. RMT is the tuple of the RMj's (recall Theorem 2.56). NMj is the index tuple of, and RMj is the internal prioritization of, the j^{th} module pre-order.

I.e., the global prioritization RG is just the result of composing the modules' prioritizations:

$$H = (RG{*}DG)$$

where

$$RG = (RE{\circ}RMT)$$

Proof : Immediate as a special case of Theorem 2.56. □

Open Default As a Module:

Recall Theorem 2.47: there one can view a predicate pre-order or a default pre-order as a parallel module. The sub-modules are the default instances' pre-orders.

2.10 Why This Definition?

In this section, we try to give some justifications for why we like the generalized definition of prioritization presented in this thesis. While it subsumes the previous definition [Lifschitz, 1985], one can imagine that there might be other definitions that do so, say, for at least the class of all finite prioritization partial orders. What's special about ours? We give evidence of two kinds.

Firstly, our generalized definition of prioritization unifies a number of previous ideas about precedence in the non-monotonic reasoning literature: we show in the next section (2.11) that those of [Geffner, 1992] and [Przymusinski, 1988], as well as [Brown and Shoham, 1989], [McCarthy, 1986] and [Lifschitz, 1985], are all special cases of our definition. Elsewhere, we show that the ideas of precedence in chronological minimization [Shoham, 1988] and persistence circumscription [Kautz, 1986] are also special cases of our definition when well-founded.

Secondly, our definition satisfies some intuitive desiderata for preference-aggregation operations in general, and for prioritization operations in particular.

2.10.1 Some Desiderata Satisfied; Commentary on Doyle & Wellman

As we discussed earlier (e.g., recall page 41), prioritization can be viewed as an aggregation operation on a tuple of "starting" preference criteria that obey the pre-order property.

Definition 2.64 (Doyle & Wellman Desiderata for Aggregation)

Doyle & Wellman [1991] (section 4.2) discuss desiderata for such an aggregation operation. They give five, then show that it is impossible to satisfy them all. Next, we list them, re-stating in our terminology and notation. Below, for brevity, we use $\preceq_R$ to stand for the aggregated preference criterion, and $\preceq_i$ to stand for the i^{th} starting preference criterion. N, as usual, is the index tuple of starting preference criteria.

1. "Collective Rationality": $\preceq_R$ is a function of the individual $\preceq_i$'s, which are unrestricted pre-orders.

2. "Independence of Irrelevant Alternatives": For any particular comparison, say between Z and Z', $\preceq_R$'s answer depends only on what the answers of the starting criteria are on that comparison (not on what they are for any other comparisons).

3. "Non-Dictatorship": There is no $i \in N$ such that: $\preceq_R$ is equivalent to $\preceq_i$ for all comparisons, regardless of the other $\preceq_j$'s answers, for $j \neq i$.

4. "Pareto Principle (Unanimity)": The global order agrees with unanimous strict preferences. In our terms:

$$(\forall i \in N.\ Z \prec_i Z') \quad \supset \quad Z \prec_R Z'$$

5. "Conflict Resolution": if Z and Z' are comparable[43] with respect to any individual $\preceq_i$, then they are comparable with respect to $\preceq_R$.

Observation: Our definition of prioritization satisfies all of these desiderata, except for "conflict resolution".

Critique of "Conflict Resolution":
In our view, "conflict resolution" is much too strong a requirement. Essentially, it rules out unresolved conflict between defaults, in the sense of conflict used in the NMR literature (see Definition 2.28). In effect, it requires all NMR to be (maximally) "credulous", rather than "skeptical".[44]

Remember, Doyle & Wellman show that at least one of their desiderata (1–5) has to be sacrificed: it is impossible to satisfy them all. Except for "conflict resolution", we agree with all of their desiderata. In our view, "conflict resolution" is the appropriate choice for what to give up. Given that, our definition of prioritization satisfies as many of their desiderata as is possible. Doyle & Wellman consider partial conflict resolution[45] and show that a prioritization-type mechanism is, in a particular sense they define, the *only* way to resolve conflict without sacrificing their other desiderata (1–4).

Definition 2.65 (Further Desiderata for Preference Aggregation)
Next, we define some further desiderata in a similar spirit to the above, that we regard as desirable for a general preference-aggregation operation.

6. "Non-strict Consensus": We would call Doyle & Wellman's "Pareto Principle": "strict consensus", since it talks about unanimity in the case of strict preferences. Another criterion is unanimity in the case of non-strict preferences: i.e., that

$$(\forall i \in N.\ Z \preceq_i Z') \quad \supset \quad Z \preceq_R Z'$$

A consequence of non-strict consensus is "tie consensus": unanimous ties results in a tie for the aggregation.

7. "Incomparability Consensus": unanimity about incomparability.

[43] i.e., not incomparable; see Definition 2.49

[44] Recall the discussion of credulous versus skeptical on page 23.

[45] Historical Note: They did so largely due to our urging in private communication at the KR89 conference.

8. "Relevance Rights": Every starting pre-order is relevant. This is a generalization of non-dictatorship. I.e., every voter's vote matters (regardless of the prioritization), in the sense that it may, under some circumstances, decide the issue. More formally: For every i and R (and N), there exists a tuple of starting pre-orders and a pair of comparisons, such that $\preceq_R$ is equivalent to $\preceq_i$. Ditto, such that $\prec_R$ is equivalent to $\prec_i$. A special case of this property[46], satisfied by our definition of prioritization, is that: If each of the starting criteria non-strictly prefers one alternative to a second, and at least one of the criteria strictly prefers the first, then the result of the prioritization strictly prefers the first. This is a kind of generalization of strict consensus.

Observation: Our definition of prioritization satisfies all of these additional desiderata (7–9) for preference-aggregation operations.

Definition 2.66 (Desiderata for Prioritization, Specifically)
In addition, one might ask: what are appropriate desiderata for a definition of a specifically *prioritization* operation, that go beyond desiderata for general aggregation operations? We can think of at least one:

10. "Pairwise Behavior": If all the criteria (voters) tie except for two that have strict priority between them, then the prioritized result is equivalent to the 2-ary case of prioritization, i.e., to the lexicographic ordering of those two criteria. The 2-ary case is the most basic and appears to be the one about which intuitive consensus is most available within the KR community. Formally:

$$\forall Z, Z'. \forall i, j \in N.\ R(j,i) \quad \supset$$

$$\{(\forall k \in (N - \{i,j\}).\ Z \approx_k Z') \ \supset$$

$$[Z \preceq_R Z' \ \equiv \ (Z \preceq_j Z' \ \wedge \ ((Z \approx_j Z') \supset (Z \preceq_i Z')))]\}$$

where $N - \{i, j\}$ stands for N minus i and j.

Observation: It is straightforward to check that our definition of prioritization indeed satisfies "pairwise behavior".

2.11 Discussion, Related Work, and Future Work

2.11.1 Inheritance and Specificity; Geffner

Many non-monotonic logical formalisms besides circumscription have some analogue of prioritization. One direction of work employs specificity as an *implicit* basis for precedence. Examples of

[46] given "non-strict consensus"

such approaches include inheritance, e.g., cf. [Touretzky, 1986] [Stein, 1989] [Geffner, 1992], conditional logics, e.g., cf. [Delgrande, 1987a] [Delgrande, 1987b], and argument systems, e.g., cf. [Loui, 1987].

One especially interesting relevant piece of work is [Geffner, 1992]. He interprets defaults as conditional assertions, and develops a theoretical treatment that employs a formalization of precedence based, like ours, on arbitrary well-founded (strict) precedence partial orders. The model preference criterion that [Przymusinski, 1988] uses to define "perfect" models (as the most preferred with respect to that criterion) is, essentially, a special case of [Geffner, 1992]'s definition.[47]

Next, we show that the formalizations of precedence in [Geffner, 1992] and, therefore, in [Przymusinski, 1988], are equivalent to a special case of our definition of prioritization. [48] We reformulate Geffner's precedence more abstractly: as a special case of an alternative definition of prioritization. Then we show that alternative to be equivalent to our definition.

Definition 2.67 (Geffner's Precedence Preference Criterion)

Geffner [1992] defines an aggregated preference criterion over the domain of models, in the context of default reasoning with precedence. Defaults are propositional (i.e., sentential; see proof below) in nature. A model is described in terms of which "assumptions", i.e., instances of default sentences, are "violated", i.e., have their negations satisfied, in that model. The precedence partial order is defined over the assumptions. Let ${\prec_R}^{\mathbf{G}}$ stand for Geffner's strict aggregated preference relation, where the precedence partial order is R. Let $M1$ and $M2$ be two (of his) models. Let $M1 - M2$ stand for the set difference of the violated assumptions in $M1$ and $M2$: i.e., those violated in $M1$ but not in $M2$. Let $M2 - M1$ be defined similarly, as vice versa. Then he defines

$$M1 {\prec_R}^{\mathbf{G}} M2 \stackrel{\text{def}}{\equiv}$$

$$\forall a \in (M1 - M2).\ \exists b \in (M2 - M1).\ R(b, a)$$

Theorem 2.68 (Geffner's Precedence is Special Case of Prioritization)

Geffner's definition of precedence corresponds to a (proper) special case of our definition (2.33) of prioritization: essentially, propositional defaults with a well-founded prioritization p.o.. [49]

Proof : The starting pre-orders are propositional-default pre-orders[50]: one per default sentence, i.e., "assumption".[51] By "propositional" here, we mean that each default formula is closed,

[47]This observation is due to Geffner (private communication).

[48]Our definition of prioritization, presented in this thesis, was developed independently of their work, at or before the same time as Geffner's.

[49]Geffner does not develop any concept of decomposing prioritizations or partial orders.

[50]Recall Definition 2.42.

[51]Technically, Geffner defines models in a slightly non-standard way, but that is not essential to this discussion.

i.e., a sentence. Let N denote the index tuple of the assumptions, i.e., of the propositional default pre-orders. Let $\preceq_i$ stand for the i^{th} such starting pre-order. Then we can reformulate: [52]

$$M1 {\preceq_R}^{\mathbf{G}} M2 \;\equiv$$

$$\forall i \in N.\, M1 \succ_i M2 \;\supset\; [\exists j \in N.\, R(j,i) \wedge M1 \prec_j M2] \tag{1}$$

Because each default ("assumption") is propositional, models are always comparable with respect to $\preceq_i$: either the assumption is true, or it is false, in each model. Thus

$$M1 \succ_i M2 \;\equiv\; \neg(M1 \preceq_i M2) \tag{2}$$

(2) implies that (1) is equivalent to:

$$\forall i \in N.\, \neg(M1 \preceq_i M2) \;\supset\; [\exists j \in N.\, R(j,i) \wedge M1 \prec_j M2] \tag{3}$$

$$\Longleftrightarrow$$

$$\forall i \in N.\, [\forall j \in N.\, R(j,i) \supset \neg(M1 \prec_j M2)] \;\supset\; (M1 \preceq_i M2) \tag{4}$$

By Theorem 2.69 (see below), (4) is equivalent to the result of prioritizing according to our definition of prioritization, i.e., Definition 2.33. **QED** □

Theorem 2.69 (Equivalence of Definition #2 of Prioritization)
Let R be well-founded. Then

$$Z \preceq_{(H;R)} Z' \;\equiv\; \forall i \in N.\, [\forall j \in N.\, R(j,i) \supset \neg(Z \prec_{Hj} Z')] \;\supset\; (Z \preceq_{Hi} Z')$$

One can read the right hand side as: "each voter votes non-strictly yes if her vote is not obviated by a strict yes vote by some individual higher priority voter".

Proof Overview: Complicated, inductive on R. See Appendix. □

Geffner [1992] shows, in a precise and general fashion, how to interpret specificity dominance as an implicit specification of precedence. Since, as we just showed, his formalization of precedence is a special case of our definition of prioritization. Therefore:

Observation 2.70 (Specificity Dominance Is Implicit Prioritization)
Specificity dominance (as [Geffner, 1992] discusses it) implicitly specifies prioritization, in a precise sense.

[52] Actually, Geffner does not explicitly define the "tie" case, i.e., $M1 {\approx_R}^{\mathbf{G}} M2$. Here, we have extended his definition (doing no violence to its spirit, we believe), to define that to be equivalent to $M1 \approx_R M2$ according to our definition (2.33) of prioritization. The latter, by Lemma B.10, is equivalent to a "tie" on each starting pre-order, and thus also is equivalent to $V[M1] = V[M2]$.

2.11.2 Model Preference Logics; Brown & Shoham

Brown & Shoham [1989] define the ideas of "biased logic" and "stratified logic". In our terms, a biased logic is defined via the set of most-preferred models of some standard "base" logic (monotonic: classical or modal) with respect to some particular arbitrary *pre-order* that is defined over the domain of such models. General circumscription, when the syntactically-defined pre-order has a well-defined corresponding model preference relation, is thus a special case of biased logic. A stratified logic is defined as a biased logic with, in our terms, a model preference pre-order that is defined as the total prioritization of some finite set of starting model preference pre-orders.[53] We defined a more general class of prioritizations, however. Thus our results show immediately how to generalize their idea of a stratified logic.

Definition 2.71 (Prioritized Logic)
We define a *prioritized logic* to be a biased logic with a model preference pre-order formed via the arbitrary, well-founded (not necessarily total or finite) prioritization of some set of starting preference pre-orders. Theorem 2.34 guarantees well-definition.

This illustrates how *our idea of prioritization is not tied to circumscription*. Our idea of prioritization is also applicable to other approaches to non-monotonic reasoning based on model preference, including [Selman and Kautz, 1989b] [Boddy *et al.*, 1989].

2.11.3 Other Approaches to Precedence

Some other NM formalisms that explicitly define partial orders of degrees of reliability, intuitively about defaults, include [Brewka, 1989a] [Ginsberg, 1988] [Zadrozny, 1987] [Pollock, 1987] [Konolige, 1988b]. Each behaves, in general, differently from our definition of prioritization, and differently from each other.[54]

Of these, **Brewka's** is perhaps the closest to prioritized circumscription. [55] He adds precedence to Poole's system [1988] of default reasoning. Poole's system is equivalent to a special case of Reiter's Default Logic: normal defaults without pre-requisites. It is also very close to a special case of parallel default circumscription (as we define it): but it is the "credulous" version (see section 8.5). Brewka defines precedence as an arbitrary finite strict partial order. Precedence acts as an ordering constraint on extension construction: in effect, it is defined proof-theoretically. Extensions are permitted if their construction corresponds to a total order on the defaults that is consistent with the partial order's constraints.[56] In section 8.8, we compare Brewka's system in further detail to prioritized default circumscription. We show there that, for a given set of default

[53]Recall Definition 2.53 of "total". Note that they use "stratified" in a sense other than what others, e.g., Lifschitz have; this motivates us to adopt the more unambiguous terminology of "layered".

[54]Our definition of prioritization was developed independently of all of these: see Historical Note in footnote at the beginning of section 2.7.

[55]Brewka's and our notions were developed independently; ours was developed in 1986 and presented publicly at the Stanford University Non-Monotonic Reasoning Seminar in spring of 1987.

[56]Brewka's system was developed independently of our definition of prioritization, at about the same time.

formulas and a given precedence ordering, his NM theories (taking the "skeptical" intersection of all the "credulous" extensions) differ in some cases from the corresponding prioritized default circumscription.

Ginsberg [1988] defines a formalism based on a bilattice of truth values. His k direction in the bilattice corresponds to confidence, and is akin to prioritization. However, since lattices are a special case of partial orders, his approach is fundamentally unable to represent many prioritization partial orders that arise naturally, such as the dags based on specificity in "multiply connected" inheritance networks, e.g., Figure 2.5.[57]

Zadrozny [1987] defines defaults as occupying a relatively permanent "referential level". A base-level reasoner attempting to construct a more transient default using a specific "object-level" theory. In doing so, it "refers" to the defaults as background knowledge in order to eliminate logically possible but less plausible interpretations. He defines "plausibility" as a strict partial order, similar to our notion of precedence and prioritization.

Pollock [1987] uses real numbers as weights, to define precedence in an argument system. Hence his basic idea of precedence is cardinal and total, rather than ordinal and partial. However, in private communication, he has pointed out that he can approximate the effect of a partial order of precedence via the partial order of counterargument-to-argument refutation relationships: one can view a weightier counterargument as having precedence over just the argument that it refutes.

Konolige [1988b] defines a "Hierarchical" variant of Moore's Autoepistemic Logic. For precedence, he permits any well-founded strict partial order. His system has a large, and, in our view, highly undesirable difference from prioritized circumscription and all of the other NM precedence formalisms we have discussed above: it rules out the existence of unresolved conflict between defaults (in the sense of Definition 2.28. As he remarks, the Quaker-Republican example (1.4) where no precedence information is available, results in inconsistency of the NM theory. As he remarks: "The HAEL structure must be constructed so that conflicts ... are avoided." One problem with this is that it requires the specifier of a NM theory to anticipate all possible conflicts — which requires *lots* of inference. A second problem is that it prohibits representing most incomplete states of default belief (knowledge). Yet the ability to represent incomplete information is much of the point of using default theories in the first place, and of permitting precedence orders to be partial.

2.11.4 General Points

Why Technical Emphasis on Pre-Orders?:
The fact that we define and discuss prioritization in terms of *pre-orders* (rather than concepts such as predicates and defaults that are perhaps more familiar to you, and to the NMR community) is important for several reasons. Firstly, the pre-order orientation enables us to unify and generalize the ideas of precedence in several NMR approaches other than circumscription, as well as in different

[57]Technically, Ginsberg is able to represent a class of precedence partial orders a bit more general than the class of lattices: namely, lattices composed with empty partial orders. This is because many default beliefs may be assigned the same truth value. Thus he is able to represent layered precedence partial orders, for example.

variants (e.g., predicate and "formula") of circumscription.

Groups of Defaults and Composition of Prioritization:
A second reason why we formulate prioritization in terms of pre-orders is that this enables us to define prioritization over *groups* of predicates and defaults, and to define *composing* prioritization. We gave motivating examples and argued that groups and composition are natural concepts that make possible the hierarchical and modular *specification* of default and NM theories. Later, we will use composition and groups to develop many of our results about inference and belief revision in circumscriptive theories. Prioritization over groups of defaults turns out to be a key basis for decomposability of inference and "bloc" safeties of updating.

Explicitness in Specifying Precedence:
The ability to specify prioritization *explicitly*, rather than implicitly as in the approaches based on specificity (e.g., in inheritance) or on recursive depth in programming languages (e.g., in logic programs with stratified semantics for negation), is important for two main reasons. Firstly, there are other bases for prioritization information which are more naturally explicit: e.g., reliability and authority. Secondly, large-scale NMR will often require the integration of different kinds of default and NM reasoning, involving multiple bases for prioritization. Explicitness is useful theoretically, as well as practically, to provide a basis for such integration.

Prioritization's Role in NMR:
Why use prioritization versus one of the other schemes to represent explicit or implicit precedence? The ultimate test is usefulness in application.

More generally, [Doyle and Wellman, 1991] suggest that there is a large space of possible methods to aggregate preference criteria for non-monotonic reasoning, and that different methods, e.g., majority voting, may be appropriate in different circumstances. Our perspective is that prioritization is just one tool in the box.

2.11.5 Future Work

An interesting direction for future work is to investigate further the relationship between specificity, e.g., in inheritance, and prioritization. Geffner's work [1992] is especially promising as a point of departure.

Our concept of prioritization applies beyond the realm of non-monotonic reasoning, to the aggregation of any preference criteria as long as they are pre-orders. One interesting potential direction of application is to formalizing social choice theory and ordinal utility as developed in economics, e.g., in [Arrow, 1963].

In [Grosof, 1992a], we further generalize our idea of prioritization to represent "pointwise" priorities cf. [Lifschitz, 1987b] [Lifschitz, 1988a]. Elsewhere (work in preparation), we still further generalize our idea of prioritization to the case of non-well-founded, infinite prioritization partial orders, e.g., to represent chronological minimization [Shoham, 1988] and persistence circumscription [Kautz, 1986].

In [Grosof, 1992a], we also show how to axiomatize, and infer, circumscriptive prioritization within the same logical language as the defaults, cf. [Lifschitz, 1987b] [Lifschitz, 1988a], and how to represent dynamically evolving, i.e., defeasible, prioritization within circumscription.

2.12 Summary

We identified a number of bases for prioritization information. These included two not previously much discussed in the NMR literature: reliability and authority. These bases motivated us to generalize the previous idea of prioritization (due to McCarthy and Lifschitz) in several ways. One is to permit *non-layered* (non-"stratified") prioritization partial orders, of a kind that arise naturally in many applications: for example, from the specificity dominance principle in inheritance, as well as from reliability or authority of sources. We showed how to generalize the prioritization partial order to be *any well-founded*, e.g., finite, (strict) partial order.

Secondly, both McCarthy and Lifschitz defined prioritization only in the context of minimizing *individual predicates* in circumscription. We showed how to *prioritize arbitrary model-preference criteria, and their syntactic correspondents*. One application of this was to generalize prioritized circumscription. We also developed the concept of *composing* prioritization and showed how to prioritize preference criteria expressing internally-prioritized *groups* of predicates (or defaults). We showed how this enables one to express hierarchical modularity in the specification of default reasoning. Our definition of prioritization is applicable to any kind of model-preference criteria as long as they obey the properties of reflexivity and transitivity. This enabled us to generalize [Brown and Shoham, 1989]'s idea of "stratified logics" to a more general category, which we dubbed "prioritized logics".

Our generalized definition of prioritization unifies a number of previous ideas about precedence in the non-monotonic reasoning literature: those of [Geffner, 1992] and [Przymusinski, 1988], as well as [Brown and Shoham, 1989], [McCarthy, 1986] and [Lifschitz, 1985], are all special cases of our definition. Elsewhere, we show that the ideas of precedence in chronological minimization [Shoham, 1988] and persistence circumscription [Kautz, 1986] are also special cases of our definition when well-founded.

In a meditative mode, we gave a number of intuitive desiderata for preference-aggregation and prioritization operations, extending those of [Doyle and Wellman, 1991]. We gave additional justification for our definition of prioritization by showing that it satisfies them.

Chapter 3

Defining Circumscriptive Theories

In this chapter, we discuss a variety of issues that arise in defining NM theories in circumscription. Our main focus is to define a new NM formalism, the Circumscriptive Logic of Defaults (CLD). This is essentially a meta-linguistic convention for specifying circumscriptive theories. This NM formalism meets three of our desires that previous variants of circumscription did not. Firstly, defaults and priorities can be specified directly as axioms. Secondly, the policy[1] parameter in circumscription is effectively standardized, making the formalism truly a single NM logical system. Thirdly, the prioritization class is more general, building on the results of chapter 2.

In addition, we give new results about "abnormality" theories, satisfiability, and fixing. In particular, we show that CLD with fixing is expressively reducible to the predicate case (of prioritized circumscription). We show that well-behavior, including satisfiability, is guaranteed for CLD theories and predicate circumscriptions with non-layered prioritization, whenever they are either universal or "quasi-"propositional (obey weak domain closure). These results underpin the applicability of our new formalism, and are used in our analyses in later chapters.

Guide to Reader: Section 3.1 provides a summary / overview of the whole chapter.

3.1 Introduction and Summary

In section 1.6.3, we discussed several expressive goals for a NM formalism, motivated by the need for expressive generality in applications:

1. fairly arbitrary partial orders of precedence;

2. updating with new defaults as a central mode of reasoning;

3. fairly arbitrary first-order forms of default beliefs, as well as of for-sure beliefs; and

4. ability to specify precedence explicitly.

[1] see Definition 2.8

In chapter 2, we showed how to meet the first of these goals: in circumscription, by generalizing prioritization. In this chapter, we will show how to meet the rest: also in circumscription, by directly specifying defaults and prioritization via axioms. We will discuss a variety of issues involved in defining circumscriptive theories of the kind we will be interested in during the rest of this paper. Essentially, we will be interested in specifying prioritized default circumscriptions (Definition 2.43) to express applications. However, some of our attention will be directed to special-case forms of prioritized default circumscriptions (e.g., with abnormality predicates and fixed formulas) that will be important later to perform our analysis: e.g., to study decomposition and reasoning procedures. Along the way, we will give new results, as well as new definitions.

In section 1.5, we introduced an informal notation for base, default, and precedence axioms. In section 3.2, we formalize this notation and extend it a bit: as a meta-language (the Circumscriptive Language of Defaults, or CLD for short) for specifying prioritized default circumscriptions.

In section 3.3, we show that prioritized *default* circumscription **expressively reduces** (is expressively essentially equivalent) to a special case: "**abnormality-style**" prioritized *predicate* circumscription. This reduction has the advantage that the predicate case has been previously much more studied in the circumscription literature, and there exist several (previous) inference procedures for it (see section 7.4). Compared to previous work on abnormalities: we formalize the usual convention of usage, and show that the folkloric expectations underlying that usage of abnormalities are borne out when there is non-layered prioritization.

Using CLD as a formalism for specification, and prioritized default circumscription as a tool for analysis, offer several advantages over using prioritized predicate circumscription. Firstly, it is less expressively constrained at surface level, as well as more natural: abnormality predicates are usually artificial ontologically. Secondly, it shows more clearly the intuitive relationship to other formalisms for non-monotonic reasoning: many, e.g., Default Logic, use a concept of a default. Thirdly, CLD offers a well-defined notion of updating the axiom set, including defaults and priorities (i.e., the policy). Fourthly, some of our analytic results about updating and inference (e.g., all of those in section 5.12, sub-section 5.13.3, and sections 6.4 through 6.6) are more naturally expressed in terms of default formulas than in terms of abnormality predicates.

In section 3.4, we show that prioritized default circumscription is **satisfiable** when: 1) the base sentence is satisfiable; and 2) either universality (e.g., clausality) or "quasi-propositionality" holds. Quasi-propositionality is a weakened domain closure condition. We observe that universality and quasi-propositionality are commonly-met conditions in AI Knowledge Representation. Previous work on satisfiability had found universality to be a sufficient condition, but had only dealt with the layered case of prioritization. Quasi-propositionality is new with us.

In section 3.5, we study the **effects and sources of fixing**. Fixing (introduced in subsection 2.3.1) is important in the previous circumscription literature as a dimension of control in defining circumscriptions. It is sometimes useful in specification; however, to what extent is still unclear. We will find fixing important for another reason, however: analytically. Fixing arises in our later decomposition results; it is closely related to monotonicity behavior in inference and updating, and to prioritization. In this section, we show that when circumscription meets the above satisfiability conditions, then the **immunity** intuition is borne out: that no new non-monotonic conclusions can be made about fixed sentences (e.g., sentences purely in fixed symbols). We also show that fixing is **expressively reducible** or inessential, in the sense that fixing a predicate or formula is equivalent to a pair of defaults. Accordingly, we extend CLD to permit axioms that specify fixing of predicates or elementary formulas. We also show how fixing of formulas can arise **indirect**ly, e.g., from fixing of other symbols or formulas, or from extreme conflict between defaults. Finally, we note that it is currently unclear whether more fixing helps, or hurts, the computational complexity of inference, e.g., in current query-answering procedures for circumscription.

3.2 Expressing Defaults and Priorities As Axioms in Circumscription

Circumscription[2] is not, strictly speaking, a non-monotonic logical formalism. Rather it forms a family of non-monotonic logics, parameterized by the circumscriptive policy. Given a particular constant policy, the circumscriptive theory is a (logically) non-monotonic function of the (for-sure) axiom set. I.e., circumscription, given the policy, is a non-monotonic operator. However, defaults are part of the policy, and we want to specify defaults as axioms: to treat them as first-class explicit information (belief) that may be updated. Indeed, we observed (section 1.6) that in many applications, much, if not most, of the updating is with default, rather than hard (non-defeasible), information. Therefore, we will develop a convenient formalism (CLD) for specifying circumscriptive theories that does directly express defaults (and their precedences) as axioms. CLD is a single, un-parametrized, NM logical formalism.[3]

Guide to Reader: The gist of the remainder of the section is: defining expression (*CLD_to_PDC*) on page 78 and the following page or so that explains it.

3.2.1 A Circumscriptive Logic of Defaults (CLD)

Definition 3.1 (Circumscriptive Logic of Defaults (CLD))
In CLD, four kinds of axioms are permitted. We saw three of these kinds in the Quaker-and-Republicans example of section 1.5, and in the Commuting example of section 1.6.2. A **base** axiom contains a prefix $\bullet>$, followed by a **sentence part**, which may be any closed first-order formula.

[2]general circumscription cf. Definition 2.7 [Lifschitz, 1984] (as well as [McCarthy, 1980] and [McCarthy, 1986])

[3]A Historical Note: Our definition of CLD developed in 1986, first presented publicly at the Stanford University Non-Monotonic Reasoning Seminar in spring of 1987, and first published in February 1989 in the IBM Research Report version of [Grosof and Russell, 1990], which also included the definition of non-layered-prioritized default pre-orders (discussed in section 2.7). CLD was also employed earlier in [Russell and Grosof, 1987].

[4] E.g.,

$\bullet\!> \forall t.\, leak(t) \supset \neg carOK(t)$

$\bullet\!> Quaker(Nixon)$

A base axiom's meaning is to assert its sentence part with for-sure (non-retractable, ordinary "monotonic") status.

A **default** axiom contains a prefix :> , followed by a **formula part**, which may be any (open or closed) first-order formula. E.g.,

$(d3) :> \ carOK(t) \wedge wayOK(t) \ \supset \ succeed(Plan1, t)$

$(d5) :> \ \neg leak(Today)$

Intuitively, a default axiom's meaning is to assert its formula part as a premise default belief. Formally, a default axiom's meaning is to specify a default pre-order corresponding to its formula part. In addition, each default axiom may, optionally, have a **label** prefix, which is a tag for specifying prioritization information via prioritization axioms. We **assume** that labels are distinct. [5]

When the formula part is open (e.g., in the first default axiom above), the default axiom can intuitively be viewed as a schema: the collection of its instantiations. This is similar to a normal default rule, without prerequisite, in Default Logic. See Observation 8.1 in section 8.5, and [Lifschitz, 1990b], for precise formalizations of this view.

A **prioritization** axiom has the form $\mathcal{PREFER}(dg, dh)$, where dg and dh are default axiom labels. E.g.,

$\mathcal{PREFER}(d5, d4)$

$\mathcal{PREFER}(d4, d1)$

A prioritization axiom's meaning is to assert that the default axiom corresponding to the first label dg has strictly greater precedence than the default axiom corresponding to the second label dh.

[4]Our restriction to first-order form for base and default axioms is not essential; however, it keeps matters simpler and suffices for the applications that we are interested in for now. We make an **exception to the first-order restriction on base axioms**, moreover. When considering serial decompositions and series circumscriptions (defined in sub-section 4.5.4, and investigated in chapters 5 through 7), we will permit the base to be itself the result of a circumscription, and thus to be second-order.

[5]This assumption is inessential and can easily be relaxed, but that would make our definitions and notation more complex.

Below, we explain how this relates to the prioritization partial order.

Terminology: Core CLD: The first three kinds of axioms form **core** CLD.

The fourth kind of axiom is a **fixture** axiom. Fixture axioms are expressively inessential: they are reducible to default axioms. We discuss them in section 3.5.

A core CLD axiom set[6] $\mathcal{A}$ specifies a prioritized default circumscription (cf. Definition 2.43) of the form[7]

$$\mathcal{PDC}(\mathcal{A}) \stackrel{\text{def}}{\equiv} PDC(B; D; R; Z) \qquad \text{(CLD\_to\_PDC)}$$

Here, as usual, Z is the tuple of all mentioned predicate symbols. By "mentioned", we mean appearing in the base and default axioms. E.g., in the Commuting example (section 1.6.2), Z is

$$\langle carOK, wayOK, succeed, leak, accident\rangle$$

B is the conjunction of the sentence parts of all of the base axioms. E.g., in the Commuting example, B is

$$[\forall t.\, leak(t) \supset \neg carOK(t)] \;\wedge\; [\forall t.\, accident(t) \supset \neg wayOK(t)]$$

D is the tuple of the default axioms' formula parts. E.g., in the Commuting example, D is

$$\langle carOK(t), wayOK(t), [carOK(t) \wedge wayOK(t) \supset succeed(Plan1, t)], \\ leak(Today), \neg leak(Today)\rangle$$

(We omit λ notation.) We let N stand for the index tuple of D: it is just (isomorphic to) the tuple of the labels of the default axioms, when indeed all those axioms have labels. E.g., in the Commuting example, N is

$$\langle d1, d2, d3, d4, d5\rangle$$

The prioritization partial order R is the transitive closure of the binary precedence relation (let us call it S) that is specified by the pairwise comparisons in the prioritization axioms. Its domain, accordingly, is the set of default axiom labels N. E.g., in the Commuting example, S contains exactly the pairs corresponding to the *arcs* in Figure 1.2:

$$(d4, d1) \quad (d5, d4) \quad (d6, d2)$$

R there contains exactly the pairs corresponding to the *paths* in Figure 1.2:

$$(d4, d1) \quad (d5, d4) \quad (d5, d1) \quad (d6, d2)$$

Thus, in the Commuting example, the maximized prioritized default pre-order is:

$$Z \preceq_{(D;R)} Z' \stackrel{\text{def}}{\equiv}$$

[6]In this paper, we consider only finite CLD axiom sets.

[7]One could instead define CLD as specifying an abnormality-style prioritized *predicate* circumscription, as we will show in section 3.3. (By "essentially equivalently", we mean up to conservative extension, cf. Theorem 3.8 and its Corollary 3.46).

$$
\begin{aligned}
&[((\neg leak(Today) \equiv \neg leak'(Today)) \;\wedge\; (leak(Today) \equiv leak'(Today))) \;\supset \\
&\qquad\qquad (\forall t.\; carOK(t) \supset carOK'(t))] \;\wedge \\
&[(accident(Today) \equiv accident'(Today)) \;\supset\; (\forall t.\; wayOK(t) \supset wayOK'(t))] \;\wedge \\
&[\forall t.\; (carOK(t) \wedge wayOK(t) \;\supset\; succeed(Plan1, t)) \;\supset \\
&\qquad (carOK'(t) \wedge wayOK'(t) \;\supset\; succeed'(Plan1, t))] \;\wedge \\
&[(\neg leak(Today) \equiv \neg leak'(Today)) \;\supset\; (leak(Today) \supset leak'(Today))] \;\wedge \\
&[\neg leak(Today) \supset \neg leak'(Today)] \;\wedge \\
&[accident(Today) \supset accident'(Today)]
\end{aligned}
$$

Note that defaults specified by default axioms without labels are in parallel with all other defaults.

We **assume** $\mathcal{A}$ is such that S is acyclic. This ensures that R is irreflexive, and thus qualifies as a strict partial order. Since the axiom set $\mathcal{A}$ is finite, N is also finite, therefore R is finite, therefore R is well-founded.

Observation 3.2 (CLD Is Well-Defined and Complete)
CLD is thus well-defined as a meta-language in the sense that every CLD axiom set specifies (maps to) a unique and legitimate (well-defined) prioritized default circumscription. Conversely, every prioritized default circumscription corresponds (maps) to a CLD axiom set. (This inverse mapping is not unique. To make it canonical: choose to include just one base axiom with sentence part B; and choose to include one prioritization axiom for each pair in the binary relation R.) Thus, as a specification tool, CLD is complete over the class of all prioritized default circumscriptions.

Definition 3.3 (CLD Theory)
We define the CLD *theory* $\mathcal{C}(\mathcal{A})$ for a given axiom set $\mathcal{A}$ to be the corresponding prioritized default circumscriptive theory (cf. Definition 2.13), i.e., the set of all conclusions entailed (model-theoretically, in second-order logic) by the prioritized default circumscription $\mathcal{PDC}(\mathcal{A})$. **Notation:** We let $\mathcal{C}$ denote the non-monotonic theory operator for the CLD formalism.

Predicate Circumscription Is A Special Case of CLD:
Note that our definition can express minimizing predicates as a special case: e.g., $:> \neg ab5(x)$. Thus (prioritized) predicate circumscription is a special case of CLD.

Definition 3.4 (Prioritized Database)
In studying inference and updating for a learning agent, we are interested in a "working" set of conclusions that is only a finite subset of the overall set of all entailed conclusions (i.e., of the CLD theory). Accordingly, we define a *prioritized database* (PDB) to be a pair, consisting of: a CLD axiom set $\mathcal{A}$; and an associated *prioritized database theory* $\mathcal{DB}$, which is some subset of the CLD theory $\mathcal{C}(\mathcal{A})$

Future Work: Expressive Generalization of CLD:
Elswhere (work in preparation), we generalize CLD expressively in several directions, building on the expressive generalizations of prioritization in [Grosof, 1992a] that we discussed in sub-section 2.11.5.

3.3 Abnormality Theories: Predicates Vs. Defaults

Guide to Reader: The gist of the section is: Definition 3.5, Theorem 3.8, and Corollary 3.15.

Typical in the applications to date of circumscription to default reasoning is the use of **abnormality** predicates, in the spirit introduced by John McCarthy [1986]. We can view an abnormality predicate ab_i as a "place-holder" for a default formula, in the spirit introduced in [Grosof, 1984] and [Lifschitz, 1984]. We saw this use earlier in the Quakers-and-Republicans example (section 2.4) and the Meetings example (section 2.8).

In this section, we formalize this use in a precise manner. Its previous treatment in the NMR literature has been rather folkloric. We show that the usual intuitions generalize to the non-layered case of prioritization; therefore, CLD and prioritized default circumscription are expressively reducible to abnormality-style prioritized predicate circumscription.

Definition 3.5 (Abnormality Theory; Abnormality-Style)
Let ab and Y be distinct tuples of predicates. Let $E[Y]$ be a tuple of (first-order) open formulas, purely in Y, similar to ab. Let $B[Y]$ be a sentence, purely in Y. We say that a prioritized predicate circumscription is an **abnormality** theory (i.e., is **abnormality-style**) if it has the **adaptive form**

$$PPC(B[Y] \wedge (E[Y] \leq ab); ab; R; Y, ab)$$

or the **definitional form**

$$PPC(B[Y] \wedge (E[Y]{=}ab); ab; R; Y, ab)$$

We will see shortly, in Theorem 3.8 and Example 3.11, the rationale for the names "definitional" and "adaptive". (On page 101, we extend these definitions to permit (a subset of the) predicate symbols to be **fixed**, as well. However, this is inessential expressively, as we will discuss later.)

Let $D[Y] \stackrel{\text{def}}{=} \neg E[Y]$. I.e., let each member of the tuple D be the negation of the corresponding member of the tuple E. Then we can write the adaptive and definitional forms above as:

$$PPC(B[Y] \wedge (\neg ab \leq D[Y]); ab; R; Y, ab)$$

and

$$PPC(B[Y] \wedge (\neg ab{=}D[Y]); ab; R; Y, ab)$$

respectively. Here, $\neg ab$ stands for the tuple each of whose members is the negation of the corresponding member of ab.

Definition 3.6 (Conservative Extension)

Let $A1[P]$ be a sentence[8] mentioning only (the tuple of symbols) P. Let Q be (a tuple of symbols) distinct from P. Let $A2[P,Q]$ be a formula mentioning only $P \cup Q$. Then we say that $A2[P,Q]$ is a **conservative extension** of $A1[P]$ when:

$$\forall P.\ [(\exists Q.\ A2[P,Q]) \ \equiv\ A1[P]]$$

or, equivalently, when both:

$$\forall P, Q.\ A2[P,Q] \ \supset\ A1[P]$$

$$\forall P.\ [A1[P] \ \supset\ (\exists Q.\ A2[P,Q])]$$

Suppose that

$$A2[P,Q] \ \stackrel{\text{def}}{\equiv}\ A1[P] \ \wedge\ U[P,Q]$$

Then we say that $U[P,Q]$ **conservatively extends** $A1[P]$, and that $U[P,Q]$ is a **conservatively extending update** to $A1[P]$. This condition is equivalent to:

$$\forall P.\ A1[P] \ \supset\ (\exists Q.\ U[P,Q])$$

Another way to view the idea of conservatism in this definition is that $A2$ "says" exactly as much about P as $A1$ does. $A2$ in addition says stuff about Q. I.e., for any formula $G[P]$ mentioning only P:

$$A2[P,Q] \models G[P] \qquad \Longleftrightarrow \qquad A1[P] \models G[P]$$

Terminology: -Free and -Part: We say that a formula G is Q**-free** when the symbols Q do not appear in G. So we can restate matters as: on the Q**-free** sub-language, $A2$ is equivalent to $A1$. I.e., the Q-free **part** of $A2$ is equivalent to $A1$. The $\exists Q$ quantifier above can be viewed as a formal mechanism to **project** the conservative extension back onto that sub-language.

Open Formulas: We extend the above definition of a conservatively extending update to permit the update U to be an open formula $U[P,Q,x]$, where x is a tuple of individual / object variables. We say that U is a conservatively extending update when its "universal closure" $\forall x.\ U[P,Q,x]$ is a conservatively extending update in the sense above.

Fact 3.7 (Explicit Definitions Are Conservative)
It is well known in classical logic that:
Explicit definitions of new symbols are always conservatively extending updates. I.e., above, let $U[P,Q]$ be a conjunction of explicit definitions of each symbol in Q:

$$\forall P, Q.\quad Q \ =\ E[P]$$

(Here, we are using the tuple $=$ notation introduced after Definition 2.4, and applying it also to functions and terms.) Then $U[P,Q]$ is a conservatively extending update.

[8] in (higher-order) classical logic

Theorem 3.8 (Conservative Extension, Abnormality Theories)
An abnormality theory in either the adaptive or the definitional forms given in Definition 3.5 is equivalent to:

$$(\neg ab{=}D[Y]) \ \wedge \ PDC(B; D; R; Y)$$

i.e., is equivalent to:

$$(ab{=}E[Y]) \ \wedge \ PFC(B; E; R; Y)$$

where D and $\neg ab$ are defined as in Definition 3.5.

In other words, the abnormality-style prioritized *predicate* circumscription is equivalent to the prioritized *default* circumscription that corresponds to maximizing $D[Y]$ (according to the prioritization R), conservatively extended by the explicit definitions of the ab predicates. Equivalently, it corresponds to minimizing $E[Y]$, in the same sense. (Recall the relationship, in Definition 2.43 of prioritized default circumscription to prioritized formula circumscription.)

Thus the ab-free conclusions of the abnormality theory (in either form) are exactly the same as those of the prioritized default circumscription.

$$\begin{aligned}
PDC(B; D; R; Y) &\equiv \exists ab.\, PPC(B[Y] \wedge (\neg ab \leq D[Y]); ab; R; Y, ab) \\
PDC(B; D; R; Y) &\equiv \exists ab.\, PPC(B[Y] \wedge (\neg ab{=}D[Y]); ab; R; Y, ab) \\
PFC(B; E; R; Y) &\equiv \exists ab.\, PPC(B[Y] \wedge (E[Y] \leq ab); ab; R; Y, ab) \\
PFC(B; E; R; Y) &\equiv \exists ab.\, PPC(B[Y] \wedge (E[Y]{=}ab); ab; R; Y, ab)
\end{aligned}$$

Here the second-order quantifier $\exists ab$ accomplishes the projection of the prioritized predicate theory, defined over the "extended" language corresponding to the symbols $\langle Y, ab\rangle$, onto the "original" (sub-)language corresponding to the symbols $\langle Y \rangle$. Thus we can regard an abnormality theory as a prioritized default theory. (On page 101, we discuss the extension of this result to the case where some predicates may be fixed. However, that is inessential expressively, as we will show later.)

Forever Equivalence: See the end of section 4.7 for an interpretation of this result in terms of "forever equivalence", under updating, between the corresponding CLD axiom sets.

Fixtures: See Corollary 3.46 for a generalization to prioritized default / predicate circumscriptions that include fixing.

Proof Overview: Non-trivial. Uses lemmas about removing second-order existential quantifiers and properties of prioritized pre-orders. See Appendix. □

Previous Results:
Theorem 3.8 generalizes Lifschitz' result ([1984], Proposition 7) [9] in which prioritization was restricted to the parallel case. This is the only previous result relating abnormalities to defaults, in circumscription, that we are aware of. Thus Theorem 3.8 is the first result that guarantees the intuitively (and folklorically) expected behavior of abnormality theories that involve non-empty, indeed non-layered, prioritization.

[9] also independently discovered by us

3.3.1 Application to Examples

Example 3.9 (Definitional Abnormality: Pacifism)
Theorem 3.8 implies that the abnormality-style formulation of the Quakers-and-Republicans example (with prioritization) in Example 2.29 is a conservative extension of the CLD formalization of that example in Example 1.5. I.e., the prioritized predicate circumscription corresponding to the former is equivalent to the prioritized default circumscription corresponding to the latter, conjoined with the explicit definition of the abnormality predicates. (In both, the prioritization partial order relation contains a single pair, corresponding to the precedence of the *Republican* default over the *Quaker* default.) In that prioritized default circumscription, the maximized prioritized default pre-order is:

$$
\begin{aligned}
& Z \preceq_{(D;R)} Z' \stackrel{\text{def}}{\equiv} \\
& \quad [\forall x.\, (Republican(x) \supset \neg Pacifist(x)) \ \supset \ (Republican'(x) \supset \neg Pacifist'(x))] \quad \wedge \\
& \quad [(\forall x.\, (Republican(x) \supset \neg Pacifist(x)) \ \equiv \ (Republican'(x) \supset \neg Pacifist'(x))) \\
& \qquad \supset \ (\forall x.\, (Quaker(x) \supset Pacifist(x)) \ \supset \ (Quaker'(x) \supset Pacifist'(x)))]
\end{aligned}
$$

Example 3.10 (Definitional Abnormality: Meetings)
Likewise, Theorem 3.8 implies that the set of *ab*-free conclusions in the abnormality-style formulation of the final variant of the Meetings example in Example 2.57 (i.e., its *ab*-free part) is exactly the CLD theory for the following CLD axiom set:

$(d1) :> \ (fire(d,t) \supset leave(p,d,t))$
$(d2) :> \ (vacat(Boss,d) \supset \neg meet(p,d,t))$
$(d3) :> \ (Tues(d) \wedge t = 4pm \supset meet(p,d,t))$
$(d4) :> \ (sick(p,d) \supset \neg work(p,d,t))$
$(d5) :> \ (holiday(d) \supset \neg work(p,d,t))$
$(d6) :> \ (weekday(d) \supset work(p,d,t))$
$\mathcal{PREFER}(d1,d2)$
$\mathcal{PREFER}(d1,d4)$
$\mathcal{PREFER}(d1,d5)$
$\mathcal{PREFER}(d2,d3)$
$\mathcal{PREFER}(d4,d6)$
$\mathcal{PREFER}(d5,d6)$
$\bullet\!> \ \forall p,d,t.\, meet(p,d,t) \supset work(p,d,t)$
$\bullet\!> \ \forall p,d,t1,t2.\, (leave(p,d,t1) \wedge (t1 \leq t2)) \supset \neg work(p,d,t2)$
$\bullet\!> \ \forall d.\, Tues(d) \supset weekday(d)$
$\bullet\!> \ Tues(Today)$
$\bullet\!> \ sick(Ed,today)$

This CLD axiom set's defaults and prioritization are illustrated in Figure 2.1.

Example 3.11 (Adaptive Abnormality; Representing Qualification)

Common in previous applications of predicate circumscription has been to write adaptive-form abnormality theories in which a base axiom[10] such as

$$\forall x. \neg ab1(x) \wedge adult(x) \ \supset \ drives(x)$$

is intended intuitively to represent a "default rule". This may, potentially, be followed later by writing another base axiom such as

$$\forall x. Manhattanite(x) \ \supset \ ab1(x)$$

which is intended to express an "exception","blocking", or "qualification" condition to the above rule. In the context of minimizing $ab1$, when there are no other axioms mentioning $ab1$, Theorem 3.8 implies that:

1. Having the first base axiom alone is equivalent, in the $ab1$-free sense, to maximizing the default pre-order corresponding to the formula

$$D1(x) \ \stackrel{\text{def}}{\equiv} \ adult(x) \supset drives(x)$$

2. Having the two base axioms together is equivalent, in the $ab1$-free sense, to maximizing the default pre-order corresponding to the formula

$$D2(x) \ \stackrel{\text{def}}{\equiv} \ \neg Manhattanite(x) \wedge (adult(x) \supset drives(x))$$

In effect, the default pre-order is **adapting** to the accumulating base axioms about $ab1$. To see the blocking behavior of (2.) relative to (1.), compare the prioritized default circumscriptions for each case, when there are no other defaults and the only base axioms are:

$$adult(Liz)$$

$$Manhattanite(Liz)$$

Then (1.) results in the non-monotonic conclusion $drives(Liz)$, while (2.) does not.

The behavior of (2.) (i.e., with blocking) is similar, but not identical (see Observation 3.12 below) to:

3. Having replaced the two above axioms about $ab1$ with the following **qualified** rule instead:

$$\neg ab1(x) \wedge adult(x) \wedge \neg Manhattanite(x) \ \supset \ drives(x)$$

which is equivalent, in the $ab1$-free sense, to maximizing the default pre-order corresponding to the formula

$$D3(x) \ \stackrel{\text{def}}{\equiv} \ adult(x) \wedge \neg Manhattanite(x) \ \supset \ drives(x)$$

[10]Here, and henceforth, we apply the terminology "base axiom", beyond CLD, to also mean a conjunct in a base sentence of some circumscription.

Observation 3.12 (Digression: Subtleties in Representing Qualification)

Next, we compare and contrast (2.) and (3.) above (in Example 3.11), i.e, $D2$, and $D3$, in more detail. Let us consider the following alternative base sentences:

$$B1 \stackrel{\text{def}}{\equiv} adult(George) \wedge \neg Manhattanite(George)$$
$$B2 \stackrel{\text{def}}{\equiv} adult(Liz) \wedge Manhattanite(Liz)$$
$$B3 \stackrel{\text{def}}{\equiv} \neg Manhattanite(Jill)$$
$$B4 \stackrel{\text{def}}{\equiv} adult(Ed)$$

Let Y stand for $\langle adult, Manhattanite, drives \rangle$.

Then (2.) and (3.) behave similarly in that:

$$PDC(B1; D2; \emptyset; Y) \models drives(George)$$
$$PDC(B1; D3; \emptyset; Y) \models drives(George)$$
$$PDC(B2; D2; \emptyset; Y) \not\models drives(Liz)$$
$$PDC(B2; D3; \emptyset; Y) \not\models drives(Liz)$$
$$PDC(B3; D2; \emptyset; Y) \models adult(Jill) \supset drives(Jill)$$
$$PDC(B3; D3; \emptyset; Y) \models adult(Jill) \supset drives(Jill)$$

But (2.) and (3.) behave differently in that:

$$PDC(B4; D2; \emptyset; Y) \models \neg Manhattanite(Ed) \wedge drives(Ed)$$

but:

$$PDC(B4; D3; \emptyset; Y) \not\models drives(Ed)$$

Rather: $PDC(B4; D3; \emptyset; Y) \models Manhattanite(Ed) \supset drives(Ed)$

Observation 3.13 (Adaptive Abnormalities: Incrementality)

The **advantage** of using adaptive abnormalities is that one can, in effect, qualify (more generally, modify) a default *incrementally* by adding new base axioms, e.g., in CLD. Indeed, one can repeatedly, in effect, modify the same default with several new base updates. E.g., in Example 3.11 above, updating the base again, with the axiom:

$$\forall x.\, imprisoned(x) \supset ab1(x)$$

results in effectively maximizing the default pre-order corresponding to the formula

$$D4(x) \stackrel{\text{def}}{\equiv} \neg imprisoned(x) \wedge \neg Manhattanite \wedge (adult(x) \supset drives(x))$$

This is an example of using adaptive-form to accumulate exception-condition qualifications to the effectively-maximized default. One exception condition is
Manhattanite; another is *imprisoned*; there might be more specified via further updates.

More generally, in a PPC with several abnormalities, each may independently be updated in this fashion to incorporate exception-condition qualifications.

Observation 3.14 (Digression: Cancelling a Default Entirely)

Note that, in like fashion, one can entirely *cancel* (i.e., in effect, retract) a default that has been represented via adaptive abnormality: e.g., by asserting the base axiom

$$\forall x.\, ab1(x)$$

in Example 3.11.

3.3.2 Expressive Reduction: Defaults to Predicates

Above, we discussed Theorem 3.8 in terms of analyzing abnormality theories. However, it can be viewed in the other direction: in terms of analyzing CLD and prioritized default circumscriptions. Theorem 3.8 tells us that:

1. Every prioritized default circumscription can be expressed, essentially, as a corresponding abnormality theory, in either adaptive or definitional form: it is equivalent to the *ab*-free part of that abnormality theory.

This tells us that the class of prioritized default circumscriptions is, essentially, expressively equivalent to the class of adaptive abnormality theories, and to the class of definitional abnormality theories.

We can view this mapping rather simply within CLD. For a given CLD axiom set $\mathcal{A}_1$, we form a correspondent CLD axiom set $\mathcal{A}_2$ that represents the equivalent (say, adaptive) abnormality theory. To arrive at $\mathcal{A}_2$, we simply *replace* each default axiom in $\mathcal{A}_1$

$$(di) :> \ Di[Z, x]$$

where $Di[Z, x]$ is a default formula open in a (possibly empty) tuple of object variables, with a *pair* of axioms:

$$(di) :> \ \neg abi(x)$$

$$\bullet\!> \ \forall x.\, \neg abi(x) \supset Di[Z, x]$$

where the abnormality predicate abi is a new symbol, and has the same arity as Di. [11] In addition, we include in $\mathcal{A}_2$ the remainder of the axioms from $\mathcal{A}_1$, all unchanged.

Next, recall that:

2. Every CLD axiom set corresponds equivalently to a prioritized default circumscription, and vice versa.

3. Prioritized predicate circumscription is just a special case of prioritized default circumscription.

4. Every abnormality theory, whether in adaptive or definitional form, is just a special case of a prioritized predicate circumscription.

[11] For definitional form, change the $\supset$ in this last, base axiom to a $\equiv$ instead.

Let CLD, PDC, PPC, $ABNAPPC$, and $ABNDPPC$ stand, respectively, for the class of CLD axiom sets, the class of prioritized default circumscriptions, the class of adaptive-form abnormality theories, and the class of definitional-form abnormality theories. Then we can summarize the above observations about expressive power with the following *expressiveness inequalities*:

$$ABNAPPC \supseteq PDC \tag{1a}$$

$$ABNDPPC \supseteq PDC \tag{1b}$$

$$CLD = PDC \tag{2}$$

$$PDC \supseteq PPC \tag{3}$$

$$PPC \supseteq ABNAPPC \tag{4a}$$

$$PPC \supseteq ABNDPPC \tag{4b}$$

Putting this all together, we have **proved** a cycle of inclusion, therefore expressive equivalence, which we can state as follows.

Corollary 3.15 (Expressive Reduction: Defaults to Predicates)

$$CLD = PDC = PPC = ABNDPPC = ABNAPPC$$

Thus prioritized *default* circumscription is an expressively inessential generalization of prioritized *predicate* circumscription, which is in turn an expressively inessential generalization of abnormality-style prioritized predicate circumscription.

This expressive reduction has the advantage that the predicate case of circumscription has been previously much more studied in the NMR literature, and there exist several (previous) inference procedures for it (see section 7.4). In sub-section 7.6.2, **we give new procedures for prioritized default circumscription that exploit this expressive reduction**.

3.4 Satisfiability and Well-Behavior

In this section, we show that prioritized default circumscription is satisfiable when (the base sentence is and) either universality (e.g., clausality) or quasi-propositionality (a generalized domain closure condition) holds. We observe these are commonly-met conditions in AI Knowledge Representation.

Guide to Reader: The gist of the section is: Theorems 3.24 and 3.28 and Observations 3.25 and 3.26.

Previous Results: Satisfiability and "Well-Foundedness":
A very basic, desirable property of a non-monotonic logical system is that the process of drawing

non-monotonic conclusions should not itself introduce inconsistency. In the context of circumscription, one would like the circumscriptive theory to be consistent, i.e., satisfiable, if the base sentence is.

Terminology: That is, one would like the circumscription operation to **preserve satisfiability**, i.e., to avoid what we call **"nasty unsatisfiability"**: the situation where the base sentence B is satisfiable, yet the circumscription sentence $C(B; policy)$ is unsatisfiable.

Etherington, Mercer, and Reiter [1985] showed that circumscription can be nastily unsatisfiable: they constructed a consistent first-order base sentence (intuitively, about the $NaturalNumber$ predicate) such that the result of minimizing a predicate ($NaturalNumber$) is inconsistent. Viewed in terms of models, the essence of the problem is that there may be no *most* preferred model because there is an *infinite* chain of models, each more preferred than the previous. This is a difficulty for any NM formalism defined (or definable) in terms of preferred models, not just for circumscription.

However, they also came up with a positive result, for the case of parallel predicate circumscription in which all non-minimized predicate symbols are fixed (**terminology:** i.e., the case with no **auxiliary predicate variables**). They identified a "well-founded-ness" condition (see Definition 3.16), on the base sentence, relative to the policy, that always implies (preservation of) satisfiability. They then showed that if the base sentence is universal (i.e., contains no existential quantifiers; see Definition 3.27), then it is "well-founded" with respect to any policy in the class. Thus for any parallel predicate circumscription without auxiliaries, if the base sentence is universal, then the circumscription preserves satisfiability. Their result was proved on the basis of ideas in [Bossu and Siegel, 1985] about a form of logical minimization different from circumscription.

Lifschitz [Lifschitz, 1986] identified a larger class of "almost-universal" base sentences, and showed that it sufficed for "well-foundedness", and thus satisfiability, for the case of parallel predicate circumscription without auxiliary predicate variables.

Unfortunately, we observe that the lack of auxiliary predicate variables is a very handicapping limitation for specification of applications of default reasoning, e.g., for learning agents.

More interestingly for our purposes, Lifschitz [1986] also generalized their notion of "well-foundedness" and their result about universal base sentences to the *layered* case of prioritized predicate circumscription *with* auxiliary predicate variables. He also showed, by example, that "well-foundedness" is not necessary for satisfiability.

Overview:
Next, we consider satisfiability for *non*-layered prioritization and default, not just predicate, circumscriptions. Clearly, it is desirable to guarantee well-behavior for common expressive classes used in AI applications. Reassuringly, we are able to show just that, for two broad cases: universal and what we call quasi-propositional.

Definition 3.16 ("Well-Founded")
We extend the previous idea of "well-foundedness" to general circumscription, and apply it to arbitrarily-prioritized default and predicate circumscription.

Let Z be a tuple of (predicate) symbols, and let H be a pre-order (alias, policy) defined over Z. We say that $B[Z]$ is *"well-founded" with respect to* H when, for every model M of B, i.e., $M \in Mod(B)$, there exists a minimal (alias, most preferred) model M' of B, i.e., $M' \in Mod(C(B; H; Z))$ that lies below it (alias, is preferred to it), i.e., such that $M' \sqsubseteq_H M$

Syntactically, we can write this as:

$$\forall Z.\, B[Z] \supset \exists Z'.\, Z' \preceq_H Z \;\wedge\; B[Z'] \wedge [\neg \exists Z''.\, B[Z''] \wedge Z'' \prec_H Z']$$

Examples: All of our Example circumscriptions in section 2.4 are "well-founded", for example; this fact follows from the last above-mentioned result in [Lifschitz, 1986].

Observation 3.17 ("Well-Founded" Implies Satisfiable)
It follows immediately from its definition that "well-foundedness" implies (preservation of) satisfiability: whenever there is a model of B, there is also a model of $C(B; H; Z)$.

Definition 3.18 ("Well-Founded" Vs. Well-Founded)
We follow [Lifschitz, 1986] in preserving the established terminology of [Etherington *et al.*, 1985]. But we use scare quotes around "well-foundedness" to distinguish it from the standard notion of well-foundedness. A partial order is **well-founded** when it has no infinite strictly-decreasing chains. We extend this idea to pre-orders. This is straightforward, since the strict version of a pre-order (i.e., $\prec_H$) has just the the same properties (i.e., transitive and irreflexive) as the strict version of a partial order.

The "well-foundedness" condition, however, does not guarantee the impossibility of infinite (strictly) decreasing chains of models. (It might be that every model in such a chain also lies (non-strictly) above some minimal model that is not in the chain.) [12]

Observation 3.19 (Well-Founded Implies "Well-Founded")
Nevertheless, well-foundedness does imply "well-foundedness", though the converse does not hold, as we just observed.

Terminology: Well-Behavior:
We say that not only (preservation of) satisfiability, but also the stronger conditions of well-foundedness and "well-foundedness", are kinds of *well-behavior*.

The fact that finite sets are well-founded under any pre-order yields guarantees in many cases of interest. Our next results are motivated by the following three observations:

- If the base language $\mathcal{L}$ is propositional, then its space of models is finite.
- if domain closure (defined below) holds, then the space of models is also finite. We can view the case of first-order with domain closure as isomorphic to the propositional-language case.

[12] "Well-founded" is basically the same as "stoppered" cf. [Kraus *et al.*, 1990]: see sub-section 4.8.1.

- We can also view closed defaults as propositional in a sense. The space of truth assignments to a finite collection of closed defaults is finite.

Definition 3.20 (Domain Closure on a Predicate)
Recall: the standard Definition 2.23 of domain closure. More generally, we define domain closure on (i.e., relative to) a particular predicate Q of non-zero arity: ($B[Z]$ entails that)

$$\forall x.\, Q(x) \quad \supset \quad [(x = t1) \vee (x = t2) \vee \ \ldots \ \vee (x = tm)]$$

Here x stands for a tuple, of the appropriate arity for Q, of individual variables. Likewise, each ti stands for a tuple, of the same arity, of ground terms. We call the above axiom: **the domain closure sentence**.

Definition 3.21 (Domain Closure on a Formula)
More generally, we define domain closure on (i.e., relative to) an elementary formula $E[Z, x]$ of non-zero arity: ($B[Z]$ entails that)

$$\forall x.\, E[Z, x] \quad \supset \quad [(x = t1) \vee (x = t2) \vee \ \ldots \ \vee (x = tm)]$$

Here x stands for a tuple, of the appropriate arity for E, of individual variables. Likewise, each ti stands for a tuple, of the same arity, of ground terms. We call the above axiom: **the domain closure sentence**.

Definition 3.22 (Quasi-Propositional)
Let $PDC(B; D; R; fix\ F; Z)$ be a prioritized default circumscription, where the tuple D is of finite length. (We use here the generalized form of prioritized default circumscription in which predicates or elementary formulas may be fixed. See Definition 3.31.)
Let $DO \subseteq D$ stand for the tuple of the default formulas that are open, i.e., are not closed.

Suppose that the base sentence $B[Z]$ entails domain closure on each open default formula DOi. Suppose, furthermore, that each of the function symbols that appear in these domain closure sentences is fixed relative to the circumscription.

Then we say that the circumscription is *quasi-propositional*.

Discussion of Definition:
Fixed-relative (Definition 3.47) is a generalization of fixed: fixed implies fixed-relative. Fixing-relative is a kind of indirect fixing. We discuss it more in sub-section 3.5.5.

Recall: our previous discussion of fixing functions, in sub-section 2.3.2, especially Theorem 2.24 and the definitions of a complete theory of equality and of uniqueness of names.

Note that each of the following is a sufficient condition for quasi-propositionality:

1. $\mathcal{L}$ is propositional

2. all defaults are closed

3. (overall) domain closure; plus all functions are fixed

4. (overall) domain closure; plus a complete theory of equality (e.g., uniqueness of names)

5. $B[Z]$ entails domain closure on each member of P that has non-zero arity, where $P \subseteq Z$ stands for the predicate symbols appearing in DO; plus all functions are fixed

6. $B[Z]$ entails domain closure on each member of P with non-zero arity, where $P \subseteq Z$ stands for the predicate symbols appearing in DO; plus a complete theory of equality (e.g., uniqueness of names)

Note that quasi-propositionality does **not require** explicit fixing of functions, nor the Uniqueness of Names, nor a complete theory of equality.

Lemma 3.23 (Quasi-Propositional Implies: Open As Instances)
The sufficient condition in Theorem 2.47 can be generalized to quasi-propositionality (Definition 3.22). **Proof Overview:** See Appendix. □

Theorem 3.24 (Quasi-Propositional Implies Well-Behavior)
Suppose $PDC(B; D; R; fix\ F; Z)$ is quasi-propositional (Definition 3.22).
Then B is **well-founded** with respect to the prioritized default pre-order
$(D; R; fix\ F)$. Therefore, B is also **"well-founded"** with respect to $(D; R; fix\ F)$. Therefore, the circumscription preserves **satisfiability**.

Proof Overview: Using Lemma 3.23, we show that quasi-propositionality leads to the finiteness of the set of discriminations that the maximized pre-order can make. This finiteness implies well-foundedness under that pre-order. By Observations 3.19 and 3.17, it follows that the circumscription is also "well-founded" and satisfiable. See Appendix for details. □

Observation 3.25 (The Quasi-Propositional Condition Is Common)
We feel that quasi-propositionality of defaults is a reasonable condition for many applications. **In practical AI Knowledge Representation**, first-order logic is very often used not so much for its increased expressive *power* relative to propositional logic, as for its increased expressive *convenience* over propositional logic. E.g., a universally quantified rule is very often used as a schema standing for the conjunction of its ground instantiations: it is much more concise. Domain closure is a common restriction in current practical theorem-provers and inference engines, monotonic as well as non-monotonic. So is a complete theory of equality, e.g., uniqueness of names. Propositionality / closed-ness of defaults is also a common restriction in NM formalisms, as well as in current practical NM inference procedures. For example, see the procedures for circumscription discussed in section 7.4.

Observation 3.26 (Universal Is Common)

Another common expressive restriction in AI Knowledge Representation is universality, i.e., no existential quantifiers. For example, **clausal** form is universal.

Definition 3.27 (Universal Formula)
The standard notion: a *universal* formula is one of the form $\forall x.\ \theta$, where θ is quantifier-free and x is a tuple of variables. We will be interested in the case of a first-order formula, where x is a tuple of object variables. Note that a conjunction of formulas is universal iff each of the conjuncts is. A theory is universal iff each of its axioms is universal; or a bit more loosely speaking, iff it is equivalent to a theory all of whose axioms are universal. A universal formula may be closed or open, i.e., it may have free variables.

Theorem 3.28 (Universal Implies Satisfiable)
In a prioritized predicate circumscription [13], if the base is universal, then satisfiability is preserved. This generalizes Lifschitz' previous result ([1986] Corollary 4.3') which was restricted to layered priority.

In a prioritized default circumscription, if the base and the default formulas are universal, then satisfiability is preserved. [14] I.e., in a CLD axiom set [15], if all formulas part are universal, then satisfiability is preserved. **Terminology:** We call this condition on CLD/PDC/PPC: **universality**.

Proof : **I)** Consider first the predicate case: $PPC(B; P; R; Z)$.
Let $R2$ be any *total* prioritization (over the index set N of P, which must be finite) that refines / extends R, i.e., such that $R \leq R2$: there always exists one. By Corollary 4.3' of (Lifschitz 1986), which covered the layered case of priority: if $B(Z)$ has a model, then since $R2$ is layered, $PPC(B; P; R2; Z)$ also has a model. But by the monotonicity of prioritization (Theorem 2.58), since $R \leq R2$, then the models of $PPC(B; P; R; Z)$ include the models of $PPC(B; P; R2; Z)$. **QED I)**

II) The default case:
is just equivalent to that for a corresponding abnormality theory, by Theorem 3.8. **QED II)**

Matters generalize straightforwardly when fixing is permitted. □

More About Well-Behavior:
Several of our later results depend on well-behavior: "immunity" of formulas in fixed symbols to the non-monotonic effect of the circumscription operation (Theorems 3.38, 3.39, and 3.56); cumulativity (sub-section 4.8.1); and any-sequence serial decomposition (Theorem 5.35).

In addition, one of our later results, Theorem 6.18, implies well-behavior for a further case: a generalization of Lifschitz' [1985] "solitary" condition.

(Also, in the proof of Theorem 5.35 (found at the end of Appendix section E.4), we will generalize the idea of well-behavior even further, to "guarded" circumscriptions and series circumscriptions.)

[13] with fixing of predicates allowed: see Definition 3.29

[14] In the generalized form in which fixing of formulas is allowed (see Definition 3.31), the fixed formulas must also be universal.

[15] with fixture axioms allowed: see Definition 3.32

3.5 Fixing and Its Effects

In section 2.3, we introduced and generalized the idea of fixing; we also gave a new result about it: Theorem 2.24 about fixing functions indirectly.

In this section, we discuss fixing's uses and give more new results that characterize its basic effects. (We do not return to the issues of fixing functions and equality, however, since that would take us too far from our main focus in this paper.)

Guide to Reader: The gist of the section is: sub-section 3.5.1.

3.5.1 Introduction and Summary

Fixing is important, firstly, as an available dimension of control in specification of circumscriptions that has been employed historically in most of the previous circumscription literature. Still unclear, however, are just what principles ought to guide its use in specification. Without these, fixing usually feels rather artificial and "technical". Fixing does not have a natural intuition behind it the way that defaults and prioritization-type precedence do.

Secondly, for our purposes, fixing is important analytically. Fixing (of predicates and elementary formulas) is closely related to prioritization, and will arise in decomposition (e.g., canonical conjunctive and serial decomposition along prioritization, discussed in sections 5.4 and 5.5). Fixing is closely related, moreover, to partial monotonicity behavior in inference and updating, and to prioritization. It thus behooves us to understand fixing's role and effects.

In this section, we

- Define its role in prioritized predicate and default circumscription.
- Illustrate intuitions about fixing by example.
- Point out the need for caution in fixing's use in specification.
- View fixing as a constraint whose effects propagate (leading, for example, to indirect fixing); and, accordingly:
- Extend CLD to permit its specification via axioms.
- Give new results that show fixing's basic effects and sources, and formalize the intuitions; in particular:
- Show that immunity (no new NM conclusions about fixed symbols or formulas) holds for broad cases, including universality and quasi-propositionality (weak domain closure).
- Show that fixing is expressively reducible or inessential, in the sense that prioritized predicate and default circumscriptions *with* fixture can be reduced equivalently to their respective cases *without* fixture. The key is that the fixing of a predicate or formula is equivalent to a pair of (extremely conflicting) defaults. As part of this, we show that the conservative extension theorem (3.8) for abnormality theories extends straightforwardly to the case with fixtures.

- Show that fixed formulas (e.g., purely in fixed predicates and functions) are globally monotonic as base or default updates.

3.5.2 Specification and Intuitive Behavior

In our earlier definitions of prioritized predicate circumscription and prioritized default circumscription, we ignored the issue of fixing. This was OK, because, as we will show, fixing is expressively inessential (i.e., reducible) in those contexts. Next, however, we elaborate those definitions.

Definition 3.29 (Prioritized Predicate Circumscription With Fixture)
We define prioritized predicate circumscription **with fixed predicate symbols** (i.e., **with fixed predicates**) as:

$$PPC(B; P^N; R; fix\ W; Z) \stackrel{\text{def}}{\equiv} C(B; (P^N; R); fix\ W; Z)$$

where $W \subseteq Z$ is a subset of the predicate symbols in the language $\mathcal{L}$. (See Definition 2.14, on page 34, for notation on right-hand side.) I.e., more explicitly:

$$B[Z] \;\wedge\; \neg\exists Z'.\, B[Z'] \wedge (W' = W) \wedge (Z' \prec_{(P^N;R)} Z)$$

Example 3.30 (Fixing Predicates: Pacifism)
In the Quakers-and-Republicans example (2.29), suppose we add to the base sentence the following:

$$\neg Pacifist(Johnson)$$

$$Pacifist(Gandhi)$$

[16] Then the following non-monotonic conclusions are entailed:

$$\mid\!\approx\; \neg Quaker(Johnson)$$

$$\mid\!\approx\; \neg Republican(Gandhi)$$

as well as the previous non-monotonic conclusion that $\neg Pacifist(Nixon)$.

But suppose we fix the predicate $Quaker$. Then the non-monotonic inference that $\neg Pacifist(Nixon)$ remains unaffected. However, one of the non-monotonic conclusions is lost:

$$\mid\!\not\approx\; \neg Quaker(Johnson)$$

In effect, fixing the $Quaker$ predicate means that no sentence purely in that predicate can be non-monotonically inferred. (Interestingly, fixing the antecedent of the default rule blocked the contrapositive direction of inference using the default. This behavior generalizes to a substantial class of examples, but fails on others. We will not take the space to go into the limitations of this use.) The non-monotonic effect of the circumscriptive policy has been weakened. Similarly, suppose we fix also the predicate $Republican$. Then also:

$$\mid\!\not\approx\; \neg Republican(Gandhi)$$

[16] as well as including the Uniqueness of Names Axiom, *per* our assumption on page 36.

Definition 3.31 (Prioritized Default Circumscription With Fixture)
We define prioritized default circumscription **with fixed formulas** as:

$$PFC(B; E^N; R; fix\ F^{NF}; Z) \ \stackrel{\text{def}}{\equiv}\ C(B; (E^N; R); fix\ F^{NF}; Z)$$

where F^{NF} is a tuple of elementary formula pre-orders. (See Definitions 2.16 and 2.17 for notation on right-hand side.) I.e., more explicitly:

$$B[Z] \ \wedge\ \neg\exists Z'.\, B[Z'] \wedge (Z' \preceq_{(E^N;R)} Z) \wedge (F^{NF}[Z'] = F^{NF}[Z])$$

where $F^{NF}[Z]$ stands for the (tuple of) formulas.
Notation: We also write F^{NF} to stand for the *formula fixture pre-order* defined as

$$Z' \approx_{F^{NF}} Z \ \stackrel{\text{def}}{\equiv}\ (F^{NF}[Z'] = F^{NF}[Z])$$

PDC notation is extended similarly.

Note that fixing a predicate is just a special case of fixing a formula. Fixing a predicate symbol P (with arity m) is just equivalent to fixing the formula $P(x)$ (where x has arity m).

Fixture As Part of the Minimized Pre-Order:
Recall by Theorem 2.19 that we can view fixture as part of the minimized (/ maximized) pre-order, in both the predicate and the default case. Recall also (footnote page 35) that this can be viewed as prioritizing the fixture in parallel with the rest of the pre-order.

Definition 3.32 (Fixture Axioms in CLD)
We define a fourth kind of axiom in CLD. A *fixture* axiom contains a prefix $\Box\!>$, followed by a **formula part**, which may may be any (open or closed) first-order formula. E.g.,

$$\Box\!> \ Quaker(x)$$

$$\Box\!> \ vacat(Boss, Today)$$

$$\Box\!> \ elephant(x) \supset gray\_skin(x)$$

A fixture axiom's meaning is to assert its formula part as a fixed formula. We extend the definition of $\mathcal{PDC}(\mathcal{A})$ in Definition 3.1 as follows. Let F stand for the tuple of the formulas that the fixture axioms specify. Then the formula fixture pre-order F, representing the fixture of all those formulas, is conjoined to the maximized pre-order:

$$\mathcal{PDC}(\mathcal{A}) \ \stackrel{\text{def}}{\equiv}\ PDC(B; D; R; fix\ F; Z)$$

Fixture axioms express constraints on the scope of the defaults' non-monotonic effects. Intuitively, they declare that certain open or closed formulas are constrained not to be "affected" by the non-monotonic closure operation.

The Need For Caution in Fixing:
In specification, one has to be careful not to over-do fixing: intended non-monotonic conclusions

may be blocked. In Example 3.30, one may have not cared about, or even preferred not to have, the "contrapositive direction" non-monotonic conclusions that were lost when fixtures were added. But, if the $Pacifist$ predicate had been fixed, then even the conclusion that $\neg Pacifist(Nixon)$ would have been lost. The next example illustrates that the effect of fixing may be indirect, and thus hard to anticipate at an intuitive level when doing specification.

Example 3.33 (Fixing Predicates: Coups)
As another example of the need for caution in introducing fixtures, consider a small axiom set with just one default: that changes of government in democratic countries are non-violent. Also there are three base axioms: that military coups always involve (at least the threat of) violence, and yet are a kind of change of government; and that January 20, 1997 will be a change of (Federal) government in the U.S.. We can represent this as the following CLD axiom set $\mathcal{A}_1$:

$:>\ change\_of\_government(x) \supset \neg violent(x)$

$\bullet\!>\ \forall x.\, military\_coup(x) \supset violent(x)$

$\bullet\!>\ \forall x.\, military\_coup(x) \supset change\_ofgovernment(x)$

$\bullet\!>\ change\_of\_government(US\_01/20/97)$

Then the CLD theory includes the conclusion that that change of government will be non-violent.

$\mathcal{A}_1 \ \ \models\!\!\approx\ \ \neg violent(US\_01/20/97)$

Now suppose that we fix the predicate $military\_coup$. (This might be motivated, say, by the intuition that we intend our default to operate in the forward, rather than backward, direction.) We can represent this fixing by adding the fixture axiom

$\Box\!>\ \ military\_coup(x)$

to form $\mathcal{A}_2$. But then the non-monotonic conclusion about non-violence, i.e., about another predicate $violent$, is lost:

$\mathcal{A}_2 \ \ \not\models\!\!\approx\ \ \neg violent(US\_01/20/97)$

One way to see why this happens is that the second base axiom implies that for this conclusion to go through (as in $\mathcal{C}(\mathcal{A}_1)$) requires also that
$\neg military\_coup(US\_01/20/97)$ be a conclusion, as well.

3.5.3 Basic Properties

Proposition 3.34 (Fixing Equivalents: Predicates Vs. Formulas)
Fixture's indirect activity can, in part, be understood by the fact that fixing a predicate or, more generally, an elementary formula, is the same as fixing an equivalent formula. More precisely: Let $C(B; H; Z)$ be a circumscription whose minimized pre-order includes the fixture of an elementary formula $F[Z, x]$ (e.g., $ab2(x)$ or $adult(x) \wedge Manhattanite(x)$):

$$Z \preceq_H Z' \ \stackrel{\text{def}}{\equiv}\ Z \preceq_L Z' \ \wedge\ (F[Z] = F[Z'])$$

If the base sentence entails that F is always equivalent to some other formula E (e.g., $Quaker(x) \supset Pacifist(x)$):

$$B[Z] \ \models\ \forall x.\, F[Z, x] \equiv E[Z, x]$$

then the fixture of F in H may, equivalently, be replaced, or augmented, with the fixture of E:

$$C(B;H;Z) \;\equiv\; C(B;G1;Z) \qquad \text{(REPLACE)}$$

$$C(B;H;Z) \;\equiv\; C(B;G2;Z) \qquad \text{(AUGMENT)}$$

where

$$Z \preceq_{G1} Z' \;\stackrel{\text{def}}{\equiv}\; Z \preceq_L Z' \;\wedge\; (E[Z] = E[Z'])$$

$$Z \preceq_{G2} Z' \;\stackrel{\text{def}}{\equiv}\; Z \preceq_H Z' \;\wedge\; (E[Z] = E[Z'])$$

(Suppose we relax the assumption that all functions are fixed. Then a similar result holds for functions and terms. If the base implies that a function, or more generally, a term is always equal to some other term, then the fixture of first term can be replaced, or augmented, by the fixture of the second.)

Proof : Very simple: substitute. □

Fixing As More Primary Than Varying:
When a symbol (or formula) is fixed either directly by specification, or indirectly by "propagation of equivalence", it then makes no difference which other symbols are varied or fixed. But when a symbol is defined as variable, it may yet be indirectly fixed. We thus view fixing as a more primary point of departure for analysis than varying.

Proposition 3.35 (Fixing The Negation Is Equivalent)
Fixing an elementary formula F is equivalent to fixing the negation $\neg F$ of that formula. More precisely, for any B and H:

$$C(B;H;fix\ F;Z) \;\equiv\; C(B;H;fix\ \neg F;Z)$$

Proof : Very simple. □

Proposition 3.36 (Formulas in Fixed Symbols Are Fixed)
Formulas purely in fixed symbols are, in effect, fixed. More precisely: Let $C(B;H;Z)$ be a circumscription whose minimized pre-order fixes the symbols $W \subseteq Z$. Let $E[W]$ be an elementary formula purely in W. Then (AUGMENT) above holds.

Proof : Very simple. □

More generally, see the definition of "fixed relative to a circumscription", and the results about it, in sub-section 3.5.5 about indirect fixing.

Terminology: We say that an update is **anti-monotonic** when it results in a circumscription / theory that is logically weaker than the previous theory, i.e., when the new theory is implied by the previous theory.

Proposition 3.37 (Fixture Is Anti-Monotonic)
Adding fixture weakens any circumscription.
I.e., adding fixture is anti-monotonic.
Let F and H be arbitrary pre-orders. Then

$$C(B;H;Z) \quad \models \quad C(B;H;fix\ F;Z)$$

In particular: in CLD, *updating with fixture axioms is anti-monotonic.*

Proof : Very simple. □

Earlier, we observed that, intuitively, fixing a predicate protects it from being affected by the minimization. Our next two results make this precise: well-behavior conditions (of the kind we studied in section 3.4) on the circumscription suffice to ensure that no new (i.e., non-monotonic) conclusion about fixed predicates will be produced by the circumscription beyond what the base sentence entails.

Theorem 3.38 (Immunity of the Fixed, Given "Well-Foundedness")
Let $B[Z]$ be "well-founded" (see Definition 3.16) with respect to a pre-order H that fixes the symbols $W \subseteq Z$:

$$Z \preceq_H Z' \quad \stackrel{\text{def}}{\equiv} \quad Z \preceq_L Z' \wedge (W = W')$$

Let E[W] be any sentence purely in the fixed symbols W. Then

$$C(B;H;Z) \models E[W] \qquad \Longleftrightarrow \qquad B[Z] \models E[W]$$

Terminology: We call this behavior **immunity of the fixed**.
Note that, in this result, we do not make the assumption[17] that all function symbols are fixed.

Proof Overview: Not terribly complicated. Uses a lemma about removal of second-order existential quantifiers. See Appendix. □
phgap

Theorem 3.39 (Immunity of the Fixed for PDC)
Any prioritized default circumscription that satisfies either of the two following well-behavior conditions also exhibits immunity of the fixed:

1. quasi-propositionality of the defaults (see Theorem 3.24)

2. universality (see Theorem 3.28)

Proof Overview: Non-trivial. Case (1.) follows immediately from Theorems 3.38 (immunity given "well-foundedness") and 3.24 ("well-foundedness" given quasi-propositionality). The proof of

[17] Recall page 36

case (2.) is similar to the proof of Theorem 3.28 (satisfiability given universality). See Appendix. □

Generalization of Immunity Results to "Fixed-Relative" Formulas:
In sub-section 3.5.5, we give a generalization of the above immunity results, to formulas that are "fixed-relative" to a circumscription: see Theorem 3.56.

Observation 3.40 (Immunity Implies Satisfiability)
Not all circumscriptions obey immunity of the fixed. In case of nasty unsatisfiability (discussed in section 3.4), any $E[W]$ is entailed by the circumscription, while only some $E[W]$'s are entailed by the base. Therefore, nasty unsatisfiability always implies violation of immunity. I.e., (preservation of) satisfiability is a necessary condition for immunity. Indeed, one can view immunity as a generalization of the satisfiability property. Satisfiability is just the special case of immunity for the formula $False$ (in which no non-fixed symbols appear).
Previous Results About Immunity of the Fixed:
The only previous results about immunity of the fixed (that we are aware of) are restricted to the case of predicate circumscription **without priorities**: Przymusinski ([1989], (Corollary 3.6)) and Gelfond *et al* ([1989] (Theorem 5.8)) showed that **universality** is a sufficient condition. [18] Our last result thus generalizes his.

Another relevant previous result for this same case, discussed in section 3.4, is that Lifschitz' almost-universal condition suffices for "well-foundedness". Theorem 3.38 **shows**, therefore, that this condition also suffices to ensure immunity of the fixed. (Lifschitz did not consider the question.)

Corollary 3.41 (Immunity of Equality When Fix All Functions)
A class of formulas in which only function symbols appear involve just equality / non-equality. Given "well-foundedness", quasi-propositionality (in a PDC), or universality (in a PDC), therefore, all equalities and inequalities between terms are immune to (i.e., unaffected by) a circumscription in which all functions are fixed.

Proof : Immediate from Theorems 3.38 and 3.39. □

Perspective on Specification: Fixing Functions:
Corollary 3.41 helps give perspective on the choice in specification to fix all functions.[19]

3.5.4 Expressive Reducibility

Lemma 3.42 (Polar Defaults)

[18] Actually, he also imposed some other conditions, e.g., about equality; those amount to having a complete, definite theory about (in)equality between all terms.

[19] Recall our assumption to that effect, on page 36.

Fixing an elementary formula $F[Z, x]$ is equivalent to a *polar* pair of defaults $F[Z, x]$ and $\neg F[Z, x]$ (with parallel prioritization between them). More precisely: any formula fixture pre-order is equivalent to the conjunction (i.e., parallel prioritization) of two default pre-orders:

$$(F[Z]{=}[Z']) \;\equiv\; \{(F[Z] \leq F[Z']) \wedge (\neg F[Z] \leq \neg F[Z'])\} \tag{PD}$$

Proof : The rightmost expression in (PD) above is just equivalent to:

$$(F[Z] \geq F[Z'])$$

□

See the next sub-section for more about Lemma 3.42's intuitive meaning and implications.

Theorem 3.43 (Reducibility of Fixture in PDC and CLD)
Fixtures are expressively inessential in prioritized default circumscription and CLD.

In terms of CLD: In any axiom set, any fixture axiom

$$\Box\!> \;\; F[Z, x]$$

can be *replaced* equivalently by the *pair* of default axioms

$$:> \; F[Z, x]$$

$$:> \; \neg F[Z, x]$$

Proof : Immediate from Lemma 3.42. □

Therefore, extending our definitions of prioritized predicate and default circumscription to permit fixtures, and extending core CLD to allow fixture axioms, do *not* affect our earlier results about expressive reducibility of CLD, PDC, and PPC to abnormality-style PPC (Corollary 3.15).

Corollary 3.44 (Previous Results: Predicate Case)
de Kleer and Konolige [1989] showed that layered-prioritized predicate circumscription with fixed predicates can, essentially, be reduced to layered-prioritized predicate circumscription without fixed predicates. "Essentially" here means that the circumscription is conservatively extended by adding the explicit definition of one new predicate $PARi$ per originally-fixed predicate PFi: that predicate is defined as the negation of the originally-fixed predicate. In the new circumscription, PF and PAR are then both minimized along with the originally-minimized predicates PN. I.e.,

$$PPC(B \wedge (PAR = \neg PF); PN; R; Y, PAR) \;\equiv$$
$$(PAR = \neg PF) \;\wedge\; PPC(B; PN; R; fix\ PF; Y)$$

(Where PN and PF are subtuples of Y.) This is, essentially, a special case of Theorem 3.43: it is proved using a special case of Lemma 3.42 and a special case of Theorem 3.8 (corresponding to [Lifschitz, 1984] (Proposition 7)). [20] It is a simple corollary of our results that their result extends to *any* R, i.e. to non-layered-prioritized predicate circumscription.

[20]We presented Lemma 3.42 and Theorem 3.43 at a Stanford University Non-Monotonic Reasoning Seminar in May of 1987, several months before de Kleer and Konolige independently made their discovery. However, we failed to publish before them.

Corollary 3.45 (Reducibility of Fixture to Predicates)

Likewise, using Proposition 3.34, it is easy to show that one can always express formula fixtures in prioritized default circumscription and CLD as predicate fixtures instead. One adds a conservatively extending update consisting of the explicit definition of new predicates, making them equivalent to the fixed formulas. Then the new predicates are fixed instead of the original formulas.

$$PDC(B \wedge (PAR = F); D; R; fix\ PAR; Y, PAR) \equiv$$
$$(PAR = F) \ \wedge \ PDC(B; D; R; fix\ F; Y)$$

Application to Immunity of the Fixed:

This result also shows how to apply our earlier results about the immunity of the fixed, which were defined in terms of fixed symbols, to the more general case of prioritized default circumscription with formula fixtures.

Corollary 3.46 (Reducing PDC with Fixtures to PPC)

Theorem 3.43 (reducibility of fixture), together with Theorem 3.8 (conservative extension for abnormalities), implies that any prioritized default circumscription with fixed (predicates or) formulas can be expressed equivalently as a conservative extension in the form of a prioritized predicate circumscription with fixed predicates in the following fashion:

$$PPC(B \wedge (ab \geq \neg D) \wedge (par = F); ab; R; fix\ par; Y, ab, par) \equiv$$
$$(ab = \neg D) \ \wedge \ (par = F) \ \wedge \ PDC(B; D; R; fix\ F; Y)$$
$$PPC(B \wedge (ab = \neg D) \wedge (par = F); ab; R; fix\ par; Y, ab, par) \equiv$$
$$(ab = \neg D) \ \wedge \ (par = F) \ \wedge \ PDC(B; D; R; fix\ F; Y)$$

where par and ab are distinct tuples of new predicates: i.e., do not appear in B, D, or F.

Terminology: We extend our definition of abnormality theory the above forms of prioritized predicate circumscription, i.e., to permit fixed predicates.

3.5.5 Indirect Fixing

Definition 3.47 (Fixed Relative to a Circumscription)

We say that a tuple of elementary *formulas* $E[Z]$ is *fixed relative* to a circumscription $C(B; H; Z)$ when

$$\forall Z, Z'.\ B[Z] \wedge B[Z'] \wedge Z' \preceq_H Z \ \ \supset \ \ E[Z] = E[Z'] \tag{1}$$

i.e., when, in the context of the base, the minimized pre-order $\preceq_H$ implies the fixture pre-order $\approx_E$ corresponding to E:

$$\forall Z, Z'.\ B[Z] \wedge B[Z'] \wedge Z' \preceq_H Z \ \ \supset \ \ Z \approx_E Z' \tag{2}$$

More generally, we will refer to any *pre-order* E satisfying (2) as fixed relative to the circumscription.

It is simple to show that the fixture pre-order corresponding to E can then, equivalently, be conjoined to the minimized pre-order H:

$$\forall Z.\, C(B; \preceq_H; Z) \;\equiv\; C(B; (\preceq_H \wedge \approx_E); Z)$$

In a circumscription: **terminology:** We can thus regard $\approx_E$ as **indirectly** fixed in the pre-order whenever it is fixed relative.

Proposition 3.48 (Base Is Fixed Relative)
In any circumscription: any sentence entailed by the base is fixed relative to that circumscription.

Proof : Very simple. □

Definition 3.49 (Forming Formulas)
We say that an elementary formula $E[Z]$ that is *formed* (purely) from the tuple of elementary formulas $G[Z]$ whenever $E[Z]$ can be built up syntactically, cf. first-order logic, from formulas $G[Z]$ using: the logical connectives $\wedge$, $\vee$, $\neg$, and first-order quantifiers (e.g., $\forall x$, $\exists y$), plus term substitution and instantiation.

Definition 3.50 (Forming Positively)
In Definition 3.49, we say that the forming is *positive* when negation ($\neg$) has *not* been used in the syntactic building.

Lemma 3.51 (Formed From Fixed Are Fixed Relative)
If the symbols W are fixed by the minimized pre-order in a circumscription, then any formula $E[W]$ purely in those symbols is fixed relative to that circumscription. More generally, relative to a circumscription: [21] If $F[Z]$ is a tuple of fixed formulas, then any formula $E[Z]$ that is formed (purely) from $F[Z]$ is fixed. E.g.:
If $E1[Z]$ is fixed , then so is $\neg E1[Z]$.
If $E1[Z]$ and $E2[Z]$ are fixed, then so are $E1[Z] \wedge E2[Z]$, $E1[Z] \vee E2[Z]$, and
$E1[Z] \supset E2[Z]$.

Proof : Very Simple. □

Another indirect source of fixtures are defaults that conflict.

Definition 3.52 (Maximal Priority)
In a prioritized pre-order $(H^N; R)$, we say that a starting pre-order Hi (or, its index i) is at *maximal* priority when there are no other starting pre-orders (indices) are at higher priority. I.e., when i is maximal (among N) with respect to the prioritization partial order R.

Note that if the prioritization is parallel, i.e., empty, then all starting pre-orders are at maximal priority.

[21] This result is dependent on our **assumption** (recall page 36) that all function symbols are fixed. If this assumption is relaxed, then restrict forming to require that substituted terms involve only fixed functions, or, more generally, only fixed terms.

Definition 3.53 (Extreme Conflict)
In a PDC: We say that two formulas $D1[Z]$ and $D2[Z]$ (e.g., default formulas) *conflict extremely* given the base $B[Z]$ when

$$B[Z] \ \models \ D1[Z] = \neg D2[Z]$$

Lemma 3.54 (Extremely Conflicting Defaults Are Fixed Relative)
In a PDC: Suppose that two default formulas $D1[Z]$ and $D2[Z]$ are at maximal priority, and conflict *extremely*. Then those two formulas are fixed relative to the circumscription. More generally, suppose that two default formulas are at maximal priority, and have partial instantiations (Definition 2.46) that conflict extremely. Then each of those two partial instantiations is fixed relative.

Proof : Similar to the proof of Lemma 3.42 about polar defaults. □

Observation 3.55 (Sources of Fixed Formulas)
To summarize: In prioritized default circumscription, fixed formulas (i.e., formulas that are fixed relative to the circumscription) may arise from:

1. explicitly specified global fixtures, e.g., fixed predicate symbols

2. base sentences and their entailments

3. a complete equality theory

4. extremely conflicting defaults at maximal priority

5. "protection" fixtures introduced as part of decomposition (see our decompositions along prioritization in chapter 5)

6. forming from (1.)–(3.)

Theorem 3.56 (Immunity of Fixed Relative)
Our earlier results showing immunity of the fixed Theorems 3.39 (universality or quasi-propositionality suffice) and 3.38 ("well-foundedness" suffices). generalize to **closed** (elementary) formulas that are fixed **relative** to the circumscription:

1. Let E be any closed formula. Then "well-foundedness" suffices for E to be immune. In particular, in PDC, quasi-propositionality suffices.

2. Let E be any quantifier-free formula. Then, in PDC, universality suffices for E to be immune.

Here, by immunity, we mean that:

$$C(B;H;Z) \models E[Z] \quad \iff \quad B[Z] \models E[Z]$$

Proof Overview: Not immediate. The proof for the "well-founded" case is similar to the proof of Theorem 3.38. By Theorem 3.24, quasi-propositionality implies "well-foundedness". For the universal case, we use Corollary 3.45. See Appendix for details. □

Example 3.57 (Indirect Fixing from Extreme Conflict, Pacifism)
Consider the conflict about Nixon in the Quaker-Republican example without priorities: Example 1.4 in CLD. Two instantiated default formulas, at maximal priority, are:

$Quaker(Nixon) \supset Pacifist(Nixon)$

and

$Republican(Nixon) \supset \neg Pacifist(Nixon)$

The base sentences $Quaker(Nixon)$ and $Republican(Nixon)$ imply that these are equivalent, in the context of the base, to:

$Pacifist(Nixon)$

and

$\neg Pacifist(Nixon)$

respectively. Thus the Quaker default and the Nixon default conflict extremely about the formula $Pacifist(Nixon)$. Thus by Lemma 3.54, that formula is fixed relative to the circumscription.

Also, in this example, the circumscription is universal. Therefore, by Theorem 3.56, immunity of the fixed holds for Nixon's Pacifism.

Suppose that one updates this example with more defaults and prioritization and fixtures (in CLD), where these updates preserve universality. Then this immunity implies that one will not conclude anything about Nixon's Pacifism *unless* the updates either:
1) assert one or more new defaults that are higher priority than the Quaker default or the Republican default; or else
2) establish a strict prioritization between the Quaker default and the Republican default.
For example, adding a new module of univeral defaults, even (in part) about Pacifism, in parallel with the Quaker and Republican defaults can never lead to a conclusion about Nixon's Pacifism.

3.5.6 Safety of Fixed Updates and Conclusions

Theorem 3.58 (Monotonicity of Fixed Base or Default Updates)
Suppose the sentence U is fixed relative to a circumscription. Then updating the base with U is globally monotonic: the circumscription after the update implies the circumscription before the update. More generally, suppose that the open formula U is fixed relative to the a circumscription. Then updating with U as a default, with any priority, is also globally monotonic.

Proof Overview: Not terribly complicated. See Appendix. $\Box$

Explicitly and indirectly fixed formulas are all safe under updates, as long as the conditions ensuring immunity are satisfied.

Corollary 3.59 (Safety of Fixed, and Indirectly Fixed, Formulas)
In CLD: Suppose that $\mathcal{A}$ is an axiom set such that $\mathcal{PDC}(\mathcal{A})$ satisfies the conditions in Theorem 3.56 that ensure immunity of the fixed: universality or quasi-propositionality or, more generally, "well-foundedness". Suppose, furthermore, that $\mathcal{U}$ is an update that preserves this sufficient condition. I.e., suppose $\mathcal{PDC}(\mathcal{A}\&\mathcal{U})$ also satisfies "well-foundedness".

Then all previous conclusions that are **fixed relative** to the previous circumscription $\mathcal{PDC}(\mathcal{A})$ are safe under the update $\mathcal{U}$. (By safe, we mean none of these conclusions are retracted after the update.)

Proof : Consider any previous conclusion E that is fixed relative to the previous circumscription. By immunity of the fixed-relative, Theorem 3.56, E must be entailed by the previous base. Since the base after the update must imply the previous base, E must be entailed by the base after the update. Therefore, E is a conclusion in the circumscription after the update. $\square$

Chapter 4

Scale: Problems, Concepts, and Strategy

In this chapter, we identify problems of scale in expressively rich NMR: in inference, in specification, and in updating, especially belief revision. The remainder of this thesis (especially, chapters 5 through 7) primarily addresses the problems of scale. We develop an analysis and strategy of attack on scale, in the spirit of divide-and-conquer. We develop concepts and motivating observations, especially about decomposition and updating, that we will use in later chapters.

Guide to Reader: Note that in the rest of this thesis, by prioritized predicate circumscription and prioritized default circumscription, we mean with fixtures cf. Definitions 3.29 and 3.31.

Guide to Reader: Section 4.1 provides a summary / overview of the whole chapter. Section 4.6 complements section 4.1: it provides a summary / overview of our strategy of attack on the problems of scale.

4.1 Introduction and Summary

In sections 4.2 through 4.4, we identify the challenge of scale in expressively rich NMR: in inference, in specification, and in updating, especially belief revision. Previous results on complexity analysis motivate a divide-and-conquer approach to these problems. We develop such a strategy of attack, at the logical level. To pursue this, however, we need a convenient idea of a part of theory. In classical logic, we take such an idea for granted. However, logical globality (sub-section 4.3.2) means we have to "work for it" in non-monotonic formalisms.

In section 4.5, we thus develop two concepts of decomposing a theory into parts: conjunctively (sub-section 4.5.3), and serially (sub-section 4.5.4). We observe that the potential gains from decomposition include locality, selectiveness, and concurrency in inference. We also observe that decomposition can potentially expose safeties of updating, by comparing a decomposition after an update to a decomposition before the update. Our concepts of decomposition are hierarchical: one can decompose into parts, then decompose those parts more finely.

In section 4.6, we recap our strategy, and supporting analysis, for attacking the problems of scale. This strategy is, to a great extent, built around exploiting decompositions.

Guide to Reader: For purposes of overviewing this chapter, section 4.6 complements this section.

In section 4.7, we introduce some basic concepts of behavior of circumscriptive theories under updating: monotonicity and redundancy. We define an interestingly stronger, "forever" version of each of these notions, as well as of equivalence. We observe that what should be an appropriate idea of equivalence between NM axiom sets, as opposed to between NM theories, is somehat problematic; and we develop a solution: "forever equivalence". Many of our results, especially in chapter 3, can be be viewed in terms of forever equivalence: for example, introduction of abnormalities.

In section 4.8, we show that the analogues of many basic properties that one takes for granted in classical monotonic logic, fail to hold in prioritized default circumscription and in many other formalisms for NM / default reasoning. This indicates that characterizing monotonicity of updating even at the level of the logic, computational challenges aside, is a tricky enterprise. One has to be careful about bringing intuitions from the monotonic case.

We discuss cumulativity in circumscription (sub-section 4.8.1) and observe the failure of rational monotony (sub-section 4.8.3) for circumscription.

Both cumulativity and rational monotony are standard concepts in the NMR literature about behavior under updating. In sub-section 4.8.2, we define a new property: "weakening for defaults". One takes for granted in classical logic that D entails any weakening $D \vee E$. However, we show that the analogue for defaults fails in circumscription.

4.2 Scale: Challenges for Non-Monotonic Reasoning

The primary issue that we address in the rest of this thesis is that of

- **Scale**: how to support non-monotonic reasoning tasks over *large* knowledge bases of rich expressive form[1].

Terminology: Tasks:
Tasks we have in mind are:

- **Specification**: the process of asserting axioms to form a particular global axiom set. We **assume** that humans are performing the specification.[2]

[1]and with strong semantics, as discussed on page 2

[2]This assumption could be relaxed. The point is that humans, at least indirectly, are the ones responsible for asserting axioms, even if only to set up the interpretation of some automatic process of information acquisition. E.g., in the commuting example of sub-section 1.6.2, humans directly create the interpretation of the reports, or else they design an automated system for perceiving and understanding the reports.

- **Inference**: the process of deriving conclusions from one particular global axiom set. (Reminder: every conclusion must be valid, relative to the non-monotonic logical system and the global axiom set.)

- **Updating**: the process of receiving a group of one or more new, asserted axioms ("**the update**"), and then modifying the stored conclusions (if any) to be in accord with the newly augmented global axiom set. This modification includes two aspects: retraction and addition. We will **assume**, for sake of simplicity of discussion, that the retractions are exactly those that are no longer valid: i.e., that the agent wishes to preserve as many of the previous (stored) conclusions as possible. (Reminder: every stored conclusion, after updating is performed, must be valid, relative to the non-monotonic logical system and the new global axiom set.) By "**belief revision**", we mean this aspect of retraction during updating.[3]

Terminology, continued:
Henceforth, when we say "**updating**", we mean receiving an update, and, sometimes (it should be clear, or else irrelevant, in context) changing the contents of the set of derived conclusions accordingly, including retaining only those previously derived beliefs that are still valid. Sometimes, we will be considering the set of derived conclusions to be the whole non-monotonic theory: this should be clear in context. E.g., when we speak of "updating the non-monotonic theory". Updating thus includes an aspect of belief revision (retraction), and also an aspect of inference (addition, and perhaps re-deriving). You may be wondering now if "inference" also includes updating: in the sense defined above, it does not. However, earlier, we sometimes used "inference" in a more general sense, to mean reasoning: this would include updating. Henceforth, we will usually be using it in the more specific sense given above; however, if it makes sense to interpret it as including updating, you should go ahead and do so. Again, this should be clear in context.

4.2.1 Previous Incapabilities

The issue of scale, for expressively rich non-monotonic reasoning (NMR) has not previously received much attention in the literature. Most of the attention it has received has been in the form of worst-case computational complexity results for expressively restricted classes, e.g. finite, propositional, and Horn (see discussion below). An exception is Rathmann [1990], who addresses issues of specification and inference for large-scale theories in a variant of circumscription. He is especially concerned with how to partition and combine theories so that they may be specified and used in pieces. He develops a framework to do so. We discuss his work in more detail in section 8.4.

[3] Because it is not our primary concern, we will not be concentrating much on the question of which new conclusions are *added* during updating, among all those derivable from the new global axiom set. E.g., we will not be concerned with whether the sentence $\neg s$ is inferred when it is derivable (valid) and s was just retracted. We only require that the additions be valid.

Current expressively rich NMR examples tend to be small: on the order of ten defaults or less. Essentially, none have moved past the "toy" stages of specificational experiment or demonstration of in-principle feasibility of inference. We would like to be able to deal with hundreds or thousands of defaults, at least.

There exist[4]:

- No forward inference procedures, except exhaustive (computes the entire non-monotonic theory).[5] (See sub-section 4.3.1 for a more precise discussion of what we mean by "exhaustive" NM inference.) No forward inference procedures at all for formalisms expressing precedence (e.g., prioritized circumscription).

- No updating procedure beyond the "brain-dead" exhaustive method of re-computing everything from scratch.

in the literature, for *any* expressively rich NM formalism[6]. Indeed, there are no such procedures even for the stratified class of PROLOG-type logic programs with negation. Nor for even the propositional / finite case of any expressively rich NM formalism (e.g., with unrestricted disjunction and negation).

Clearly, it would be desirable to achieve the capabilities to be **selective**[7] in forward reasoning, and to be selective in belief revision: as in first-order-logical inference applications. **The exhaustiveness of the previous methods is, in effect, a manifestation of their caution in dealing with interaction.** In our view, these previous incapabilities demonstrate the need to "beat globality" of interaction and context-dependence.

There has been more work on procedures for backward inference in expressively rich NM formalisms than for forward inference. [Reiter, 1980] gives a backward procedure for the "normal" class of Default Logic. Ginsberg [1988] gives a backward procedure for his Multi-Valued Logic. [Ginsberg, 1989] [Baker and Ginsberg, 1989] [Przymusinski, 1989] [Inoue and Helft, 1990] [Helft *et al.*, 1991] give backward procedures for broad cases of predicate circumscription, including many prioritized cases. These cases include (supersets of) propositional predicate circumscription without priorities, and propositional predicate circumscription with priority that is "stratified" (i.e., what we call "layered"; see Definition 2.48).[8]

But neither these nor any other inference procedures for expressively rich NMR have yet[9] been effectively implemented to develop real software applications.

[4] to our knowledge

[5] Moreover: these procedures are for finite (usually, propositional) cases only.

[6] in the sense that we defined on page 2, including circumscription, Default Logic, and Autoepistemic Logic

[7] See sub-section 4.3.1 for more precise discussion of the notion of selectiveness in NM inference.

[8] See chapter 2 for the definitions of prioritized and predicate circumscription. See sections 7.1 and 7.4 for more discussion of previous inference procedures: in circumscription.

[9] to our knowledge

4.2.2 Computational Complexity

Part of the problem is that all of the above-listed backward inference procedures appear quite computationally expensive: NP-hard, or worse, for unrestricted propositional cases; and non-semi-decidable for cases with unrestricted first-order forms of belief; where the complexity is known.

Known complexity results indicate that, even when goal-directed, default reasoning is intrinsically intractable in the worst case, for a variety of the expressively richer (cases of) formalisms [Reiter, 1980] [Kolaitis and Papadimitriou, 1988] [Selman and Kautz, 1989a] [Kautz and Selman, 1989] [Stillman, 1990] [Selman and Levesque, 1989] [Eiter and Gottlob, 1991] [Stillman, 1992] [Gottlob, 1992]. Especially interesting is the fact that skeptical entailment[10] has been recently shown to be Π_2^P-complete for the propositional case of several expressively rich non-monotonic logics, including: predicate circumscription without priorities [Eiter and Gottlob, 1991], the "normal" case of Default Logic [Stillman, 1992] [Gottlob, 1992], Autoepistemic Logic [Gottlob, 1992] [11], and several other non-monotonic modal logics [Gottlob, 1992].

Π_2^P-complete is standard terminology in the area of computational complexity theory ([Garey and Johnson, 1979], section 7.2). It stands for co-NP-complete in oracles that are themselves NP-complete. More colloquially, this is sometimes phrased as: "second level in the polynomial hierarchy", where "first level" is NP-complete (or co-NP-complete). If $P \neq NP$, then the polynomial hierarchy does not collapse, and thus Π_2^P-complete is strictly harder than NP-complete.

Recall that entailment in ordinary monotonic propositional (classical) logic is NP-complete. What the Π_2^P-completeness means is that there is an entirely different source of (co-)NP-hardness in expressively rich propositional *non*-monotonic reasoning, that goes beyond the aspect of monotonic reasoning. We can summarize this by saying **(terminology)** that:

> **The non-monotonic aspect** of reasoning is (co-)NP-hard.

Our perspective is that **conflicting interactions between defaults can be viewed as a major source of this extra computational difficulty of the non-monotonic aspect**. In the following two sections, we expand on this perspective.

Known tractable cases of default reasoning are highly restricted. [Selman and Kautz, 1989a] [Kautz and Selman, 1989] [Stillman, 1990] give polynomial-time backward procedures for special cases, including restrictions of Horn, of propositional default reasoning in Kautz and Selman's model-preference logic and in Default Logic. Delgrande [1991] gives a polynomial-time backward procedure for a Horn propositional case of his conditional logic. [Eiter and Gottlob, 1991] and [Gottlob, 1992] give and review some other cases of tractability. For example, reasoning with the Closed World Assumption is not much harder than purely monotonic reasoning (for propositional cases) [Eiter and Gottlob, 1991].

[10] Recall page 12 for the terminology "skeptical", and page 23 for discussion of skeptical versus credulous entailment.

[11] See section 8.5 for a review of Default Logic, including the definition of the "normal" case. See section 8.6 for a review of Autoepistemic Logic.

4.2.3 Historical Perspective: First-Order Logic

Of course, AI is full of worst-case intractability. The message from the above grim results about worst-cases is not to despair. Rather, it is to hunt for tractable methods, for useful restricted cases, and for ways to match them to each other.

One perspective on the history of automated inference is that, in its early phases, lots of groundwork went into the "primary" understanding of the logical system, e.g., of reformulations, equivalences, and logical implications. This underlay the subsequent development of first-order proof procedures: first, to get such at all, then to get relatively efficient ones and find special cases, e.g., Horn, that were both expressively interesting and computationally exploitable.

After early inventions such as substitution of truth values for subexpressions and exhaustive forward chaining, automated (propositional and) first-order inference really began to accelerate in effectiveness with the discovery of useful context restrictions and expressive restrictions for backward chaining (query-answering).

Here, by "context", we mean what information needs to be considered in the midst of a proof procedure: axioms and intermediate conclusions (and negated goals). By "context restrictions", we mean ways of restricting relevant context. By "expressive restrictions", we mean restrictions on the expressive form (class) of the axioms (and goals, queries, conclusions).

These useful **context restrictions for FOL** include: unifiable; set of support (in the resolution sense); and linear (in the resolution sense).

These useful **expressive restrictions for FOL** include: clausal (no existential quantifiers); Horn; and appearance of function symbols (none beyond 0-ary; finite Herbrand base). For example, Datalog[12] combines all of these expressive restrictions.

4.2.4 High-Level Strategy Motivated by Analogy

We view the previous state of historical development of automated inference[13] for expressively rich non-monotonic logical formalisms, as analogous to first-order inference just before the above kinds of discoveries of useful context restrictions and expressive restrictions.

Our overall, high-level strategy for tackling the problems of scale, therefore, is to:

- Seek better primary logical understanding of default reasoning.
- Aim to develop results about useful context restrictions and expressive restrictions.

The above-mentioned results by Kautz, Selman, Stillman, and Delgrande on tractable special cases of default reasoning represent early achievements of the NMR community toward the goal of discovering interesting expressive restrictions. Their strategy, in this regard, is similar to ours. Interestingly, their results, and those of the others who have developed procedures, show that the above-listed expressive restrictions for FOL are useful for NMR as well.

[12]Datalog is a standard concept in the logic programming literature. For example, see [Ullman, 1988], especially pages 100-106, for a description.

[13]including updating

4.3 Globality and Locality in Inference

4.3.1 Reducibility, Exhaustiveness, Selectiveness

Recall in sub-section 4.2.2, we discussed the Π_2^P-completeness of skeptical inference for the propositional case of several expressively rich NM formalisms, including parallel predicate circumscription.

We **observe** that, therefore, inference in PPC and in PDC is Π_2^P-complete: since PPC and PDC include parallel predicate circumscription as a special case.

A further implication of this computational complexity is that to reduce NM inference to monotonic inference, for PDC or for the other expressively rich NM formalisms, takes co-NP-hard computational effort.

Suppose (as many do in computer science) that NP-completeness means exponential time cost, as does co-NP-completeness, at both the first and the second level of the polynomial hierarchy. Then this suggests that one can achieve significant computational-time savings *if* one can manage to **reduce (terminology)** NM inference to monotonic inference, *without* incurring exponential work in the reduction or exponential blow-up in the size of the reduced representation. Next, we explain what we mean here by "reduce".

A circumscription is a *second*-order formula[14]. However, often, a circumscription is equivalent to a *first*-order (e.g., propositional) formula expressed in the first-order (e.g., propositional) base language $\mathcal{L}$. In this case, we can write the following equivalence:

$$C(B;H;Z) \quad \equiv \quad B[Z] \wedge GA[Z]$$

where $GA[Z]$ is first-order (e.g., propositional). In the circumscription literature, one says that GA is a **generating axiom (terminology)** for the non-monotonic conclusions / for the circumscription / for the circumscriptive theory. Note that in "generating axiom", "axiom" is meant in the monotonic-logical sense, *not* in the sense of being part of a NM axiom set.

For example, in the Quaker-Republican example with priority (Example 1.5, formalized in CLD), a generating axiom for the circumscription is:

$$\begin{aligned}&[\forall x.\, Republican(x) \ \supset\ \neg Pacifist(x)] \\ &\qquad \wedge\ [\forall x.\, Quaker(x) \wedge (x \neq Nixon) \ \supset\ Pacifist(x)]\end{aligned}$$

More generally, by "generating axioms", we mean a collection of sentences whose conjunction is a generating axiom in the above sense.

The virtue of finding a generating axiom for a circumscription is that one can then draw all the conclusions in the circumscriptive theory by performing purely monotonic inference from the base plus generating axiom. Thus finding a generating axiom for a circumscription is a reduction of the problem of NM inference in the circumscription to a problem of monotonic inference. Notice that

[14]More precisely, circumscription as we have defined it for purposes of this thesis. Recall the footnote in Definition 2.7

for the above example, there was not much blow-up in the size of the representation in moving from the CLD axiom set to the base plus generating axiom.

We say that inferential reasoning is **exhaustive in the non-monotonic aspect (terminology)** when it produces a *monotonic*-logical representation that is equivalent to the circumscription. (More briefly, we will just say **exhaustive**.) E.g., when that reasoning produces a generating axiom for the whole circumscription. Informally, this corresponds to "computing the entire non-monotonic theory" (up to monotonic inference).

We say that inferential reasoning is **selective in the non-monotonic aspect (terminology)** when it produces a *monotonic*-logical representation that is equivalent to part of the circumscription, i.e., that is implied by the circumscription (and, in turn, implies the base). (More briefly, we will just say "**selective**".) E.g., when that reasoning produces a generating axiom for some of the conclusions in the circumscriptive theory. Informally, this corresponds to "computing part of the (entire) non-monotonic theory" (up to monotonic inference). For example, this part might be defined as complete with respect to only a proper subset of the base language $\mathcal{L}$, or with respect to only a proper subset of the global set of defaults[15].

Similar notions of reduction, exhaustiveness, and selectiveness can be defined for many other NM formalisms, including skeptical Default Logic (see section 8.5) and skeptical Autoepistemic Logic (see section 8.6).

4.3.2 Non-Monotonicity is Logical Globality

Internal Updating and External Context:
Logical non-monotonicity is defined by behavior under updating with new "external" axioms. However, one can view updating as an *internal* issue as well. Relative to a part of the current axioms, the remainder of the (overall set of) current axioms can be thought of as either "*external context*", or as an "*internal update*".

Context-Dependence:
In our examples in section 1.5, we saw that, in general, in NMR, the conclusions sanctioned by a piece of information, i.e., by a subset of the axioms, depend non-monotonically on what other axioms are present, i.e., the external context. This context-dependence is qualitatively different from the situation in first-order logic, for example. There, the set of conclusions sanctioned by a subset of axioms is, essentially, unchanged upon consideration of the rest of the axioms in the global axiom set. By contrast, in NMR, soundness, not just completeness, is at risk upon consideration of the rest. (We will expand on this point when discussing the concept of compositionality in section 4.5)

Non-Modularity:
Thus another way to view non-monotonicity is as *non-modularity*. In NMR, one cannot, in general, blithely draw conclusions from a part of the overall "**global**" set of axioms, as if there were

[15] see section 5.8 for a precise notion of completeness with respect to a subset of defaults

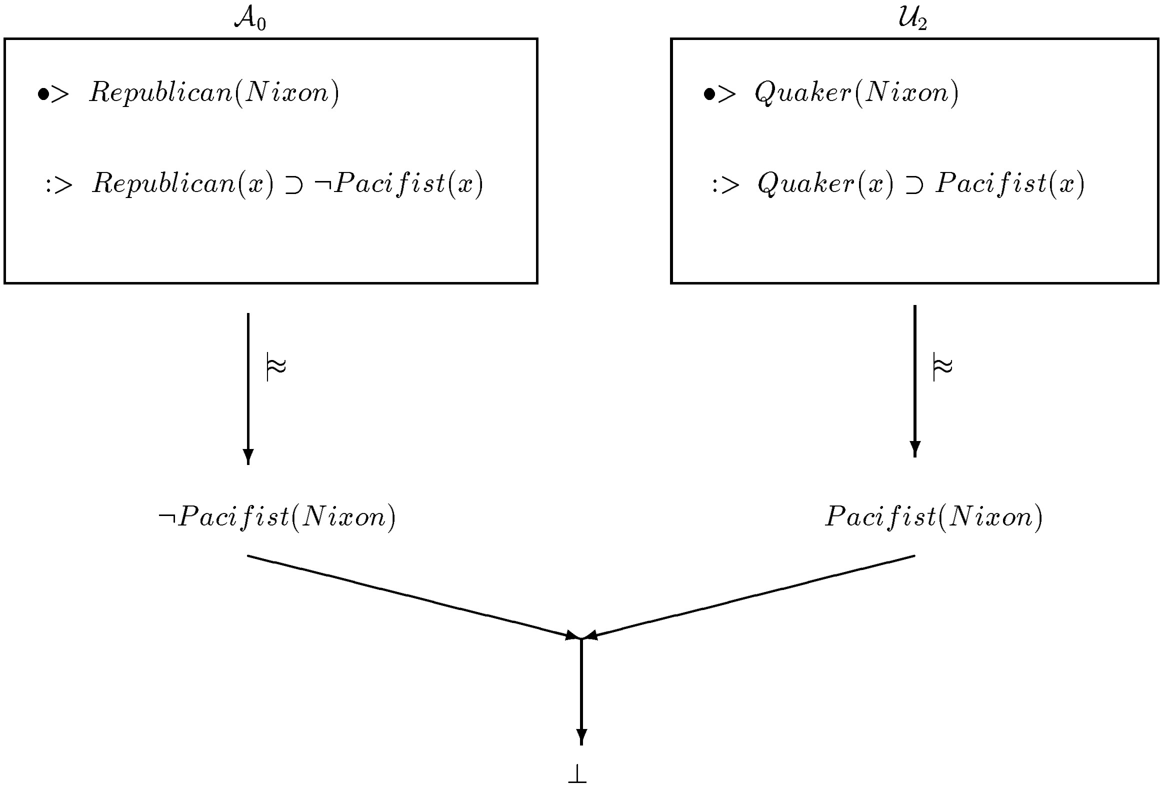


Figure 4.1: Non-modularity: Quakers and Republicans. (Default axiom labels not shown.)

no other axioms present, and expect to get the same result as by drawing conclusions from the whole. (By "same" here, we mean in the sense of soundness.) In Example 1.4, consider two parts of the global set of axioms. When we consider $\mathcal{A}_0$ as if it were a global axiom set, we conclude $\neg Pacifist(Nixon)$. When we consider $\mathcal{U}_2$ as if it were a global axiom set, we conclude the opposite: $Pacifist(Nixon)$. Figure 4.1 illustrates. Conjoining the two theories $\mathcal{NMTH}(\mathcal{A}_0)$ and $\mathcal{NMTH}(\mathcal{U}_2)$, produces "garbage": inconsistency. This is a different result than the overall non-monotonic theory $\mathcal{NMTH}(\mathcal{A}_0\&\mathcal{U}_2)$.

Entailment is Global:

This example shows that in NMR, unlike in monotonic logic, "omission is significant", in the following sense. It is the *absence* of the axioms in $\mathcal{U}_2$ that enables $Nixon$'s non-pacifism to be concluded from the axioms in $\mathcal{A}_0$. Entailment in NM logical systems is "*global*" in the sense that it is a function of the global axiom set, rather than on any "local" subset. This is due to the possibilities for conflicting interaction, e.g., between the Quaker default and the Republican default.

It seems desirable to understand the nature of context-dependence and conflicting interaction in NMR: we will be investigating it.

4.3.3 Need for Locality in Inference

Logical globality poses a severe challenge for NM inference, generally.[16] Computationally, one would like to be able to proceed with the task of NM inference in "local" fashion: to divide up the work into independent sub-tasks, hopefully concurrently. In a sense, this is necessary, not merely desirable. For large axiom sets, a computational inference procedure cannot consider all the axioms simultaneously to start with: i.e., when the size of the axiom set exceeds the single-cycle (parallel) processor capacity. Most inference procedures for monotonic logics form lemmas, or something analogous. These lemmas, or analogues, are based on limited, non-global information.

Unfortunately, as we observed in the last sub-section (4.3.2), *all* of the external context is relevant *a priori*. *Non*-monotonicity is simply the removal of the guarantees of logical monotonicity, i.e., context-independence and modularity (in the senses discussed in sub-section 4.3.2). The general case is *caveat emptor*[17].

Inference, whether backward or, especially, forward[18], involves a problem of "internal" updating. If one imagines a procedure that, in effect, draws intermediate conclusions, first doing some work with part of the axioms and later proceeding to consider more of the axioms, then, in effect, it is receiving updates.

Non-monotonicity, i.e., non-modularity, means that *a priori*, any lemma formed from a subset of the axioms may need to be revised upon considering more of the rest of the axioms. Thus to form lemmas in the first place, as well as to form lemmas concurrently, is problematic.

In this sense, NMR thus places one in a situation of
Holism that Is A Potential Computational Nightmare.

Difficulties with Rules of Inference:
Even the applicability of the concept of an inference rule, as that concept exists in monotonic logic, is made problematic by non-modularity. An inference rule such as resolution, or *modus ponens* in natural deduction, is *local* and context-independent in the sense that it has a small, fixed number of (is-provable) antecedents. In effect, it is a rule that sanctions a mini-lemma. The above point, about the need of lemmas for locality, thus applies. Some non-monotonic logical systems are even defined in terms of inference rules, e.g., Default Logic and, essentially, Prolog-style logic programs with negation. But these inference rules typically contain one or more consistency (non-provability) conditions as antecedents. These consistency conditions are fundamentally globally context-dependent. E.g., the fixed-point construction in Default Logic.
Strategy:
What is needed to guide inference is the delimitation of **relevant context (terminology)**: to be local at some scope, rather than fully global. That is, one needs to be able to pick some manageable piece of the global premise information to work with locally, in such a manner that all the rest of

[16] **Terminology Reminder**: By the task of "**inference**", we mean inference from one particular axiom set. The task of updating, by contrast, involves revision as well.

[17] Latin for "let the buyer beware", i.e., no guarantees implied or intended.

[18] see footnote definition of backward and forward on page 5

the global premise information is **irrelevant (terminology)**[19] to the question of global correctness of inference from this local information.

Ideally, one could just select *any* subset of the overall axiom set, and work with it locally. However, this is simply not possible in NM logical systems, in general. We will aim to organize inference to work with modular pieces of the overall axiom set. We will aim to understand the structure of conflicting interaction: how to view a theory as composed of **parts** (see section 4.5). We will aim to discern and to summarize relevant context.

We will focus on precedence information as one particularly useful tool to relate structure in specification to structure in interaction.

We will show how to recover, in many cases, some of the modularity that is lost in moving from monotonic to non-monotonic logic. In some special cases, this modularity is perfect: the external context is irrelevant. But more usually, it is imperfect. We will show how relevant context can often be summarized, short of including all the other axioms, in a manner analogous to the role of external declarations in programming languages with side effects.

We will aim to understand inferential opportunities for concurrency, and more generally, the constraints on sequencing, e.g., when one may validly treat an intermediate conclusion as non-retractable henceforth. Finally, we will aim to recognize relevance, modularity, and constraints on sequencing, in a way that is relatively easy computationally, e.g., on the basis of the syntax of the axioms involved.

4.3.4 Specification

Specification of non-monotonic theories, e.g., knowledge acquisition of the axioms, also shares the problem of logical globality, i.e., non-modularity.

Specifying non-monotonic theories is notoriously tricky. To write even a few axioms that correspond to intended behavior can be quite difficult. Sometimes this is due to lack of understanding of the logical system's properties. But we **observe** that much of the difficulty is due to the **human computational** problem: it is hard for people to figure out what are the entailed and non-entailed conclusions sanctioned by any "draft" axiom set. A common step in forming "draft" axiom sets is to add one or more new axioms, and then to see what changes to entailment result. The human computational problem will grow if non-monotonic logical systems are ever to be used for large knowledge bases. Specification thus inherits much of the difficulties that inference (sub-section 4.3.3) and updating (section 4.4) owe to globality and non-modularity.

Specification as a process involves a need for modularity in a more direct way, as well. If large specifications are ever to be developed, a practical necessity will be to develop them in pieces, and then to combine the pieces. This combination can be viewed, essentially, as "internal" updating. As in the structured methodologies for large-scale software development, it will be desirable to achieve as much modularity as possible in the "piece" specifications, so as to ease the job of developing

[19] For formalizations of the notion of irrelevance, see [Subramanian, 1989] and [Levy, 1992].

each. What is needed, in part, is a **structure to aid specification** that enables a "user" to delimit the scope of his or her axioms' effects on entailment. This will be especially important if several people are to co-operate on specifying a large axiom set.

Strategy:

Our strategy for tackling these problems of specification is similar to that for inference. We will aim to understand the structure of interaction, and how to summarize relevant context. We will aim to find special cases where the external context is completely irrelevant.

4.4 Belief Revision; Safety in Updating

The difficulty posed by non-modularity appears perhaps in its purest form in the task of **belief revision** during **updating**. Logical non-monotonicity, alias logical non-modularity, is defined by behavior under updating with new axioms: i.e., the potential for retraction.

As we discussed earlier in sub-section 1.6.1, we are especially interested in applications of NMR to learning agents (as defined there). Consider a learning agent that is incrementally acquiring new axioms as time goes on, and that is maintaining a stored body of derived beliefs. These derived beliefs are sound, i.e., valid inferences, with respect to the non-monotonic logical system. Each update $\mathcal{U}_i$ consists of one or more axioms. Let's call time i the moment after the i-th update. At any given time n, the agent has a collection of axioms $\mathcal{A}_n$, and a body of derived conclusions $\mathcal{T}_n$ that is some subset of $\mathcal{NMTH}(\mathcal{A}_n)$.

Next, consider the situation after another update $\mathcal{U}_{n+1}$. What happens to the previous conclusions $\mathcal{T}_n$? In general, some of them still are valid inferences from $\mathcal{U}_{n+1}$. However, some of them are not. In general, the agent must **revise** its beliefs, in the sense of retracting some that were in $\mathcal{T}_n$ but are not justified (sound, valid) to include in $\mathcal{T}_{n+1}$. (For the sake of simplicity in this discussion, let's also assume that the agent keeps all its previous conclusions that are still valid.) In addition, there may be some conclusions in $\mathcal{T}_{n+1}$ that were not in $\mathcal{T}_n$: either newly-valid inferences, or inferences that simply were not made and stored before.

Figures 4.2 and 4.3 illustrate the "before" and "after", respectively.

Logical non-monotonicity is exactly the absence of the guarantee that, after receiving new axioms (the update), any particular previous conclusion will still be sanctioned (valid). *A priori*, no potentially retractable conclusion is **safe** from the need to be revised after any particular update. Figure 4.3 illustrates this predicament intuitively: the retractions are spread "randomly" among the previous conclusions.

Terminology: Henceforth, when we say "**update**", we mean a group of one or more new axioms. When we say "**updating**", we mean receiving an update, and, sometimes (it should be clear, or else irrelevant, in context) changing the contents of the set of derived conclusions accordingly, including retaining only those previously derived beliefs that are still valid.[20] By "**belief revision**", we mean this aspect of retraction during updating.

[20]See also our definition of the task of "updating" in the beginning of section 4.2.

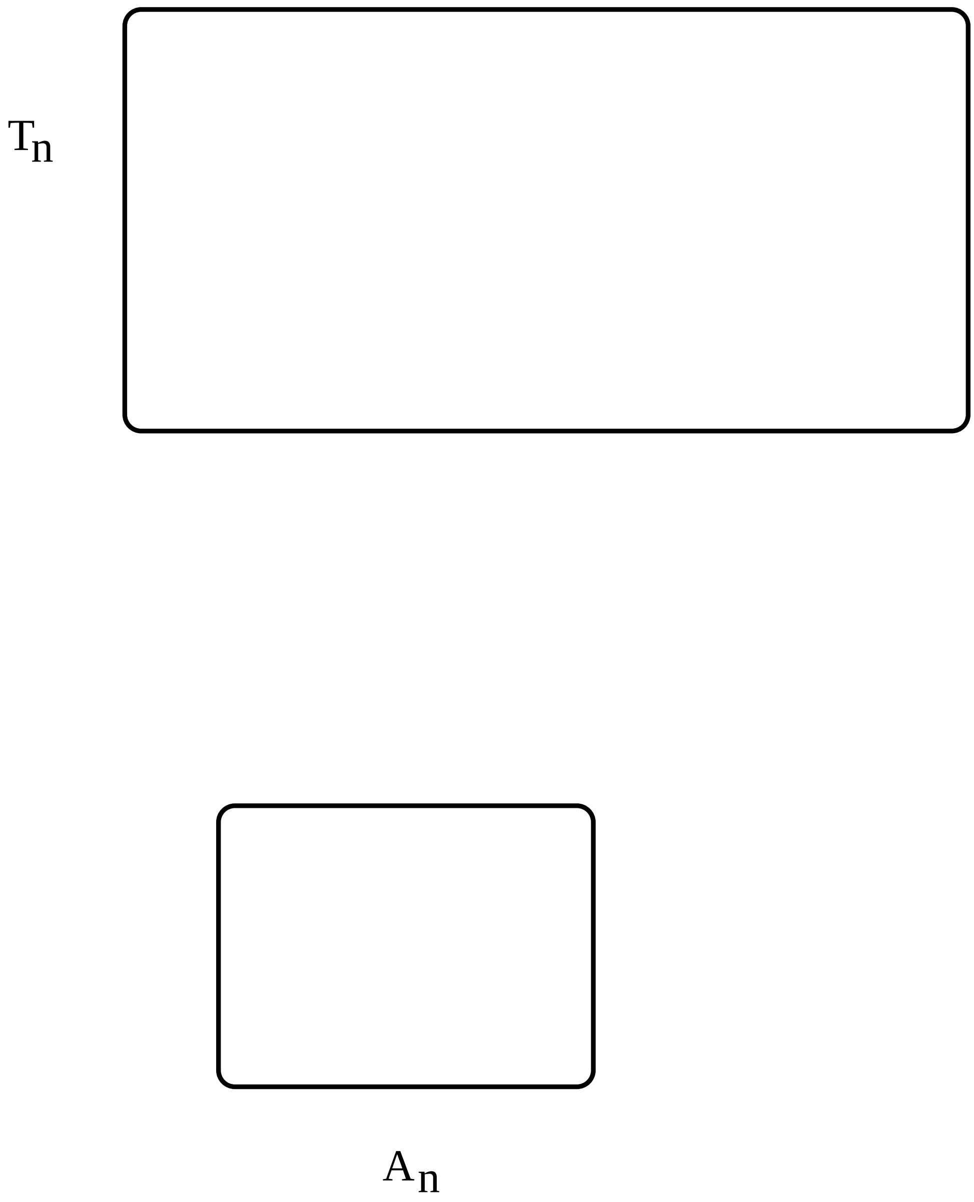


Figure 4.2: Axioms $\mathcal{A}$ and beliefs $\mathcal{T}$: BEFORE update.

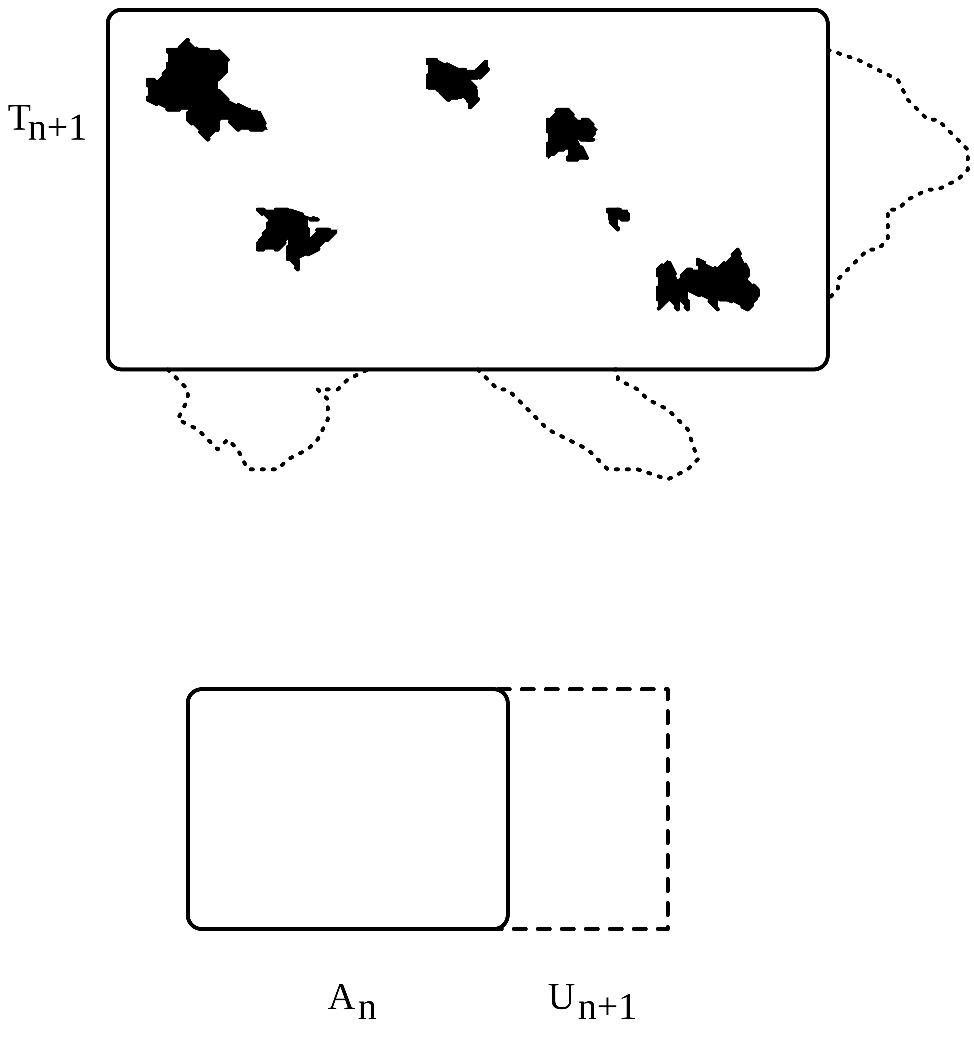


Figure 4.3: AFTER update. Blotches indicate retractions. Dotted regions indicate new conclusions. The dashed region indicates new axioms. Difficulty: retractions are spread “randomly”.

The task of belief revision during updating poses a severe **computational** challenge. In order for a (large-sized) learning agent to maintain previous conclusions during a process of updating, with any hope of tolerable computational cost, it needs to avoid having to do a **global recomputation** of all the (potentially retractable) conclusions! Humans' beliefs are highly stable: the vast majority do not change after each little piece of new information, e.g., in a continuing stream of perception and communication. Also, many of humans' most stable beliefs are the ones that took the most work to infer (e.g., scientific knowledge). We would like to be able to design automatic learning agents that emulate these properties: stability of belief, and avoidance of continually re-inferring all of one's beliefs "from scratch".

We will thus be addressing the question:

- For a given update, how can one reduce the arena of potential revision to a smaller **part** of the conclusions?

Special Role of the Belief Revision Task:
The task of belief revision manifests directly the problem of logical globality. In inference and in specification, one can try to avoid the need for retraction, which is present implicitly (as we discussed in sub-sections 4.3.3 and 4.3.4). In belief revision, however, one cannot evade actually "doing" the retractions. This is one reason we will be focusing on the belief revision task, especially.

Strategy:
We will show how to delimit the **unsafe** zone of conclusions from a **safe (terminology)** zone of conclusions where revision, and thus recomputation, is unnecessary. We will aim to reduce the unsafe zone as much as possible. Figure 4.4 illustrates this goal intuitively: we would like to partition off many previous conclusions into the "happy" safe zone.

We will aim for these safe zones to be determinable "ahead of time", with as little computational effort as possible. We will aim to organize the global knowledge base accordingly, so as to facilitate the task of updating.

Our strategy is to find **structure** in the set of axioms (old and new), and relate it to structure in the set of conclusions, in such a way that some **parts** of the previous conclusions can be known, via theorems about the logical system, to be completely safe. Figure 4.5 illustrates the overlaying of structure onto Figure 4.3: some parts of the conclusions are entirely free of retractions.

We will treat this question of safety for the entire valid set of conclusions, i.e., the **global non-monotonic theory**. Of course, the stored conclusions of a useful computational agent will usually, due to practical resource constraints, be only a fraction of the whole non-monotonic theory for expressively rich cases. And, even in principle, often they can only ever be a proper subset, due to finiteness, e.g., for expressive cases subsuming first-order logic.

We will develop a collection of theorems about what parts of the non-monotonic theory are safe in the face of particular updates: in other words, what parts of the non-monotonic theory the

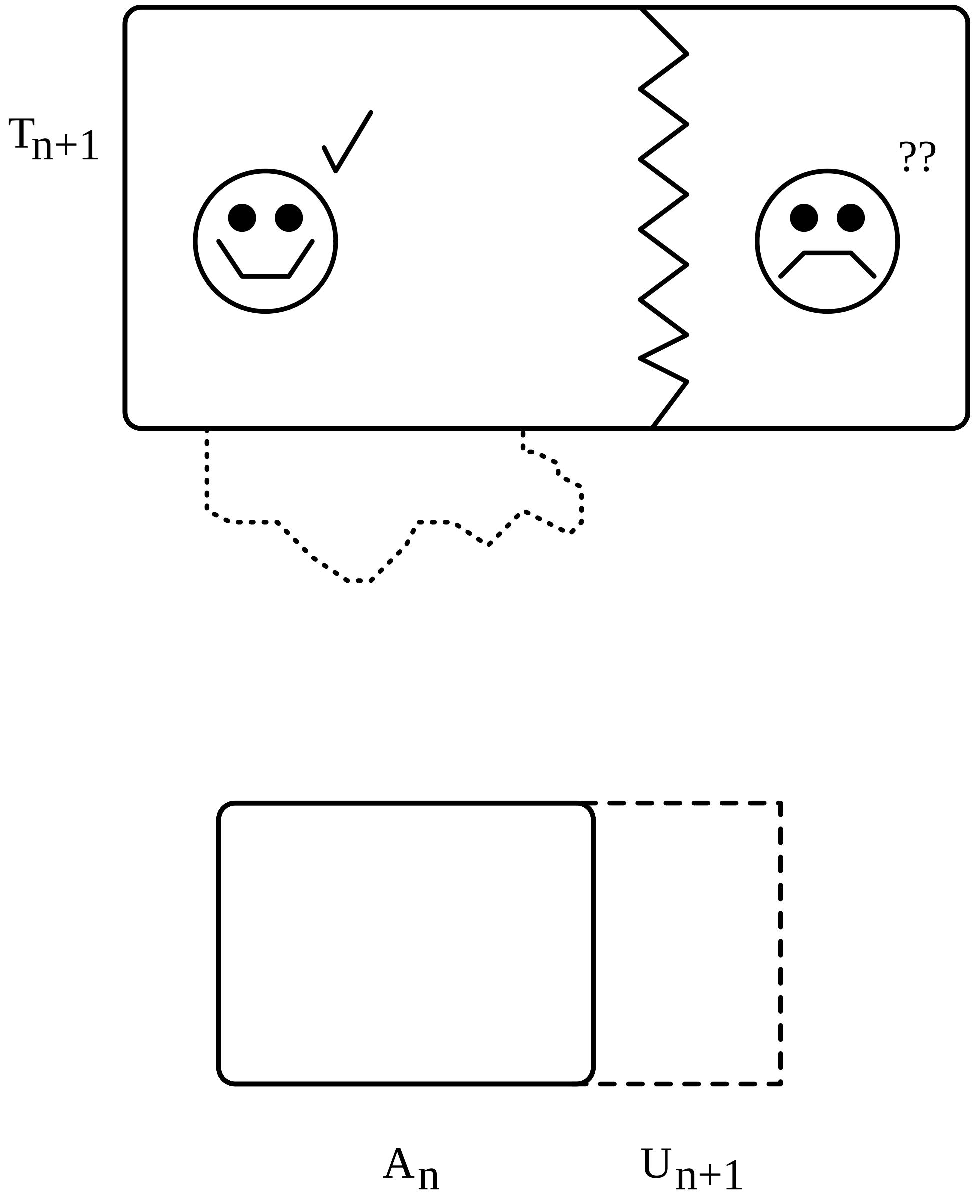
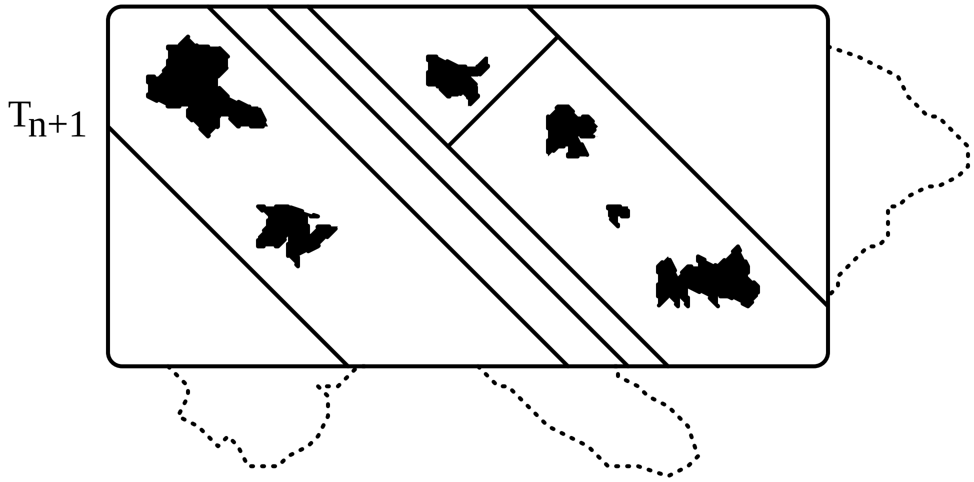


Figure 4.4: Safe (happy) and unsafe (unhappy) zones. A zone is safe when it is known to contain no retractions. Goal: to maximize partial safety, i.e. "don't worry, be happy" as much as possible.



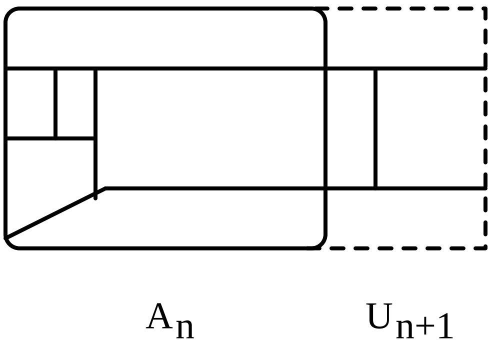


Figure 4.5: Structure creates safe zones.

update affects *monotonically*.

Terminology: We call such safety conditions **partial monotonicities of updating**.

We will also develop results about when updates are **globally monotonic (terminology)**, i.e., do not require retraction of any conclusion in the entire previous non-monotonic theory.

We will give theorems about monotonicities of updating not just for new for-sure axioms, but also for **updating with new defaults and precedences** (priorities), the importance of which we discussed in section 1.6.3.

4.5 Concepts of Theory Decomposition

In this section, we develop concepts of theory decomposition that we will use throughout the remainder of this thesis.

4.5.1 Introduction; Strategic Role

As part of our strategy, we need to develop a strong idea of a **part** of a non-monotonic theory. This is important for several reasons:

- to define safe versus unsafe zones for belief revision;
- to define relevant versus irrelevant context for inference (and for specification); and
- to define the structure and organization of an overall ("global") prioritized database (Definition 3.4), i.e., knowledge base.

In classical logic, we take for granted a convenient such idea of a part of theory. However, logical globality and non-modularity mean that we have to "work for it" in NM logical systems.

In more detail: In classical logic, the consequential closure $Cn(\mathcal{SA})$ of *any subset* $\mathcal{SA}$ of the global axiom set $\mathcal{A}$ forms a "sub-theory" that is a subset of the global theory $Cn(\mathcal{A})$. In NMR, the theory $\mathcal{C}(\mathcal{SA})$ formed from a subset $\mathcal{SA}$ of the global axiom set $\mathcal{A}$ is *not*, in general, a subset of the global theory $\mathcal{T} \stackrel{\text{def}}{\equiv} \mathcal{C}(\mathcal{A})$. The remainder $\mathcal{S} - \mathcal{SA}$ of the global axiom set is relevant context when attempting to draw NM conclusions from $\mathcal{SA}$.

We draw two different ideas of a part-to-whole relationship from first-order logic: **conjunctive decomposition** and **serial decomposition**. In first-order logic, both of these kinds of decomposition are "effortlessly" available.

In first-order logic, one can break up a global axiom set into *any* collection of subsets whose union is the global set, perform consequential closure on each subset of axioms separately, then combine the consequences conjunctively[21], and get the same set of consequences as for the global axiom set. We call this **conjunctive** decomposition: the global theory is equivalent to the conjunction of, i.e., is conjunctively composed from, a collection of sub-theories corresponding to the axiom subsets.

[21]In sub-section 4.5.3, we define this conjunctive combination more precisely.

Also in first-order logic, one can take *any* axiom subset, perform consequential closure on it, then add *any* other axiom subset to those consequences (now treating the consequences as axioms), perform consequential closure (of the two together), add *any* another axiom subset, perform consequential closure, and so on for all the axiom subsets, in the manner of a cascade. Again, if the union of the collection of axiom subsets is the global axiom set, then one gets the same set of consequences as for the global axiom set. We call this property **serial** decomposition: the global theory is equivalent to the serial consequential closure of, i.e., is serially composed from, a collection of sub-theories corresponding to the the axiom subsets.

The unconstrained conjunctive and serial decomposition available in first-order logic is a basic property underlying inference in it. By contrast, none of the non-monotonic logical systems that we have studied, including circumscription, Default Logic, Autoepistemic Logic, and inheritance systems[22], have this effortless kind of conjunctive and serial decomposition: most choices of "break-up" of a global axiom set into a bunch of axiom subsets do not produce an equivalence to the global. The Quaker-Republican example (Example 1.4 and Figure 4.1) illustrated this difficulty.

Nevertheless, in the remainder of this section, and in chapters 5 through 7, we will show how both to conjunctively decompose and to serially decompose non-monotonic theories with a fair degree of flexibility. These decompositions are less simple in structure than for first-order logic, and often apply only in special cases.

4.5.2 Formalizing Decomposition

Next, we formalize the concept of decomposition.

Definition 4.1 (Decomposition of NM Theories and Axiom Sets)
Our general concept of decomposition is applicable to many NM logical systems. A global theory $\mathcal{T}$ can be obtained *either* directly by applying the NM theory operator $\mathcal{C}$ to the global axiom set $\mathcal{A}$, *or* indirectly (but equivalently) via decomposition. In decomposition, the global axiom set $\mathcal{A}$ is *decomposed* into an associated set of *constituent* axiom sets (the $\mathcal{SA}_i$'s). The global theory $\mathcal{T}$ is then equivalent to the *combination* of the sub-theories (the $\mathcal{ST}_i$'s) that correspond to each constituent axiom set. Two kinds of combination are: conjunctive and serial. Figure 4.6 illustrates conjunctive decomposition with a flow diagram.

Intersections Among Constituent Axiom Sets:
In general, a global axiom set and theory may have many different decompositions. In each, an interesting issue is whether the constituent axiom sets for the sub-theories are actually *subsets* of the global axiom set. In general, we do *not* require them to be. We will see later that it is often interesting to consider decompositions in which each constituent axiom set is a *mutation* (**terminology**) of a (non-empty) subset of the global axiom set: the mutation being to include additional axioms that, intuitively, summarize the relevant external context. Another interesting issue is whether the axiom sets for the sub-theories are *disjoint*, or if they overlap, in what way. In

[22] see references listed on page 2

general, we do *not* require that $\mathcal{SA}_i$ be disjoint from $\mathcal{SA}_j$ for $j \neq i$. We **require** only that, for a given decomposition, one does not include the other. Later, we will see many examples of mutation and overlap in chapter 5 and of overlap in chapter 6.

4.5.3 Conjunctive Decomposition and Its Uses

We will focus more on conjunctive than on serial decomposition.

Definition 4.2 (Conjunctive Combination and Decomposition)
In conjunctive decomposition, each sub-theory is simply the result of applying $\mathcal{C}$ to the corresponding constituent axiom set:

$$\mathcal{ST}_i \stackrel{\text{def}}{\equiv} \mathcal{C}(\mathcal{SA}_i)$$

In CLD, we say that $\mathcal{T}$ is the result of *conjunctive* combination when

$$\mathcal{T} = Cn(\bigcup_{i=1,\ldots,n} \mathcal{ST}_i)$$

where Cn is the *monotonic* consequence (theory) operator in classical logic.[23] Figure 4.6 illustrates. When the corresponding axiom sets are understood, we will say that the global *theory* is conjunctively decomposable into these *slice* sub-theories.

In terms of the circumscriptions, we have:

$$\mathcal{PDC}(\mathcal{A}) \equiv \bigwedge_{i=1,\ldots,n} \mathcal{PDC}(\mathcal{SA}_i)$$

Again, when the corresponding axiom sets are understood, we will also speak of a *circumscription* being conjunctively decomposable into slice circumscriptions, e.g., for $n = 2$:

$$PDC(B; D; R; Z) \equiv PDC(SB1; SD1; SR1; Z) \wedge PDC(SB2; SD2; SR2; Z)$$

Aims in Decomposition:
What can one hope to gain from the study of decomposition? Overall, it is to divide-and-conquer the complexity of a global theory. We will aim to find decompositions for which, in some sense, each sub-theory is *simpler* than the global theory; e.g., that its axiom set is smaller or expressively less complex than the global axiom set.

Observation 4.3 (Locality in Inference)
A conjunctive decomposition, shows how one can soundly, and in an important sense completely, perform inference *locally*: considering only the axioms in $\mathcal{SA}_i$, and using whatever *standard* procedures are available for CLD (more generally, for the NM formalism of interest). This is sound, because $\mathcal{C}(\mathcal{SA}_i)$ is then a subset of the global theory. This is complete, in a sense, because the contribution of $\mathcal{SA}_i$ to the global consequences requires only *monotonic* inference beyond its own local NM consequences $\mathcal{C}(\mathcal{SA}_i)$. (Remember, the Cn operation that combines the sub-theories is just ordinary classical monotonic entailment.) In short:

[23] This could be generalized to consider other monotonic theory operators, e.g., for modal formalisms.

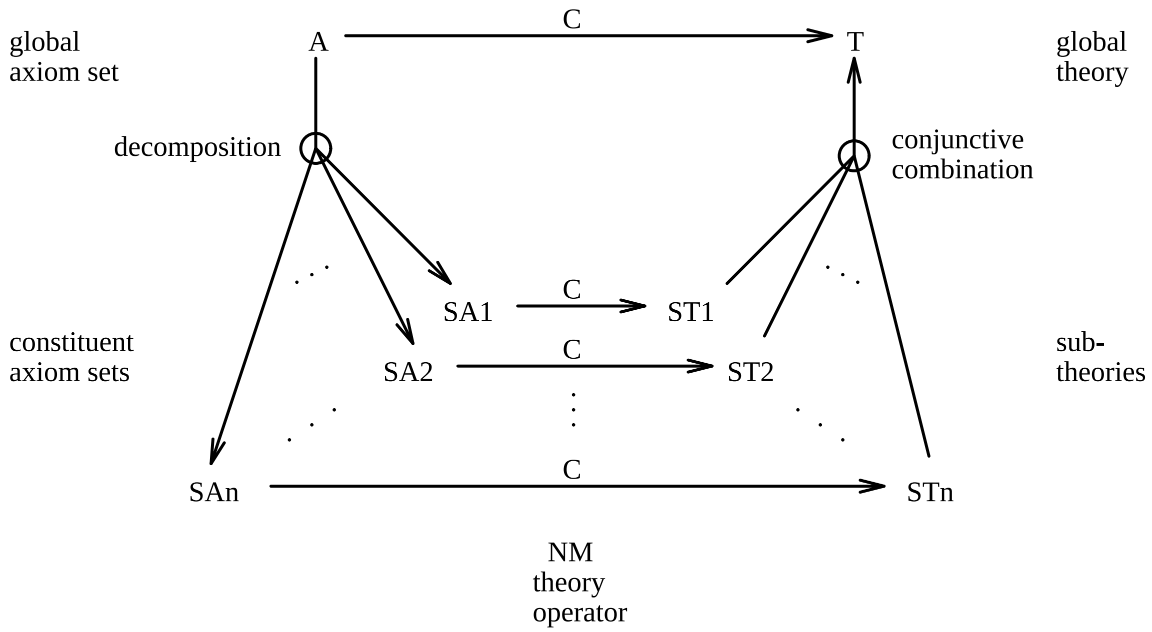


Figure 4.6: Conjunctive Decomposition: a conceptual flow diagram. A global theory $\mathcal{T}$ can be obtained *either* directly by applying the NM theory operator $\mathcal{C}$ to the global axiom set $\mathcal{A}$, *or* indirectly (but equivalently) via decomposition. In decomposition, the global axiom set $\mathcal{A}$ is *decomposed* into an associated set of *constituent* axiom sets (the $\mathcal{SA}_i$'s). The global theory $\mathcal{T}$ is then equivalent to the *conjunctive combination* of the corresponding sub-theories (the $\mathcal{ST}_i$'s), where each sub-theory is the result of applying $\mathcal{C}$ to a constituent axiom set: $\mathcal{ST}_i \stackrel{\text{def}}{=} \mathcal{C}(\mathcal{SA}_i)$.

All non-monotonic inferences can be localized to the slices; only monotonic inference is required between the slices.

Examples: canonical decomposition in sections 5.4 and 5.5, totally-prioritized modules in sub-section 5.13.1, totally-prioritized propositional defaults in sub-section 5.13.2, disjoint sub-languages in section 6.2, sympathetically solitary in section 6.5.

Observation 4.4 (Selectiveness in Forward Inference)
Conjunctive decomposition also shows how forward inference can be conducted *selectively* (i.e., non-exhaustively) with respect to the global, even when, to start with, one has only exhaustive forward inference procedures available for CLD (more generally, for the NM formalism of interest):

Exhaustive inference on a slice generates only a selective part of the global theory.

The finer the grain of the decomposition, the more selective this technique can be. See discussion of hierarchy below (Observation 4.9) for more about grain size.

Examples: canonical decomposition in sections 5.4 and 5.5, totally-prioritized modules in sub-section 5.13.1, disjoint sub-languages in section 6.2, sympathetically solitary in section 6.5.

Observation 4.5 (Concurrency in Inference)
Conjunctive decomposition exposes concurrency:

Inferences within *each slice can be performed in parallel with inference* within *every other slice.*

And remember that all of the NM inference can be contained within the slices. The NM aspect of the inference task, often the most difficult, is thus *divided* by conjunctive decomposition.

Examples: canonical conjunctive decomposition in section 5.4, disjoint sub-languages in section 6.2, sympathetically solitary in section 6.5.

Observation 4.6 (Safeties of Updating)
Conjunctive decomposition can expose safeties (partial monotonicities) of updating: by comparing a decomposition after an update to a decomposition before the update. E.g., suppose, in conjunctive decomposition, that some of the constituent axiom sets, say:

$$\{\mathcal{SA}_1, \ldots, \mathcal{SA}_k\}$$

are present both before and after an update $\mathcal{U}$. That is, suppose the previous global axiom set $\mathcal{A}$ has a conjunctive decomposition whose constituent axiom sets are:

$$\{\mathcal{SA}_1, \ldots, \mathcal{SA}_k, \mathcal{SA}_{k+1}, \ldots, \mathcal{SA}_m\}$$

and suppose that the updated global axiom set $\mathcal{A}\&\mathcal{U}$ has a conjunctive decomposition whose constituent axiom sets are:

$$\{\mathcal{SAU}_1, \ldots, \mathcal{SAU}_k, \mathcal{SAU}_{k+1}, \ldots, \mathcal{SAU}_n\}$$

, where

$$\mathcal{SA}_1 = \mathcal{SAU}_1 \quad \wedge \quad \ldots \quad \wedge \quad \mathcal{SA}_k = \mathcal{SAU}_k$$

Then we know that all of the conclusions in the conjunctive combination of their associated slices: (which we can write as)

$$\bigwedge_{i=1,\ldots,k} \mathcal{C}(\mathcal{SA}_i)$$

are safe under the update. In short:

Differential decomposition yields safeties of updating.

Examples: higher-priority conclusions in section 5.12, disjoint sub-languages in section 6.2, conservatively extending updates in section 6.4, sympathetically solitary in section 6.5, strongly sympathy in section 6.6.

Definition 4.7 (Clean Slicing)
Suppose we can find a conjunctive decomposition in which for some i, the slice's axiom set is indeed a *subset* of the global, i.e.,

$$\mathcal{SA}_i \subseteq \mathcal{A}$$

In this case, we say that the slice is a *clean* slice.

Observation 4.8 (Clean Slicing Yields Strong Locality and Irrelevance)
If a slice is clean, then we know that all the remaining axioms in the global axiom set are *irrelevant context*, in a particular sense, relative to the slice's axiom set. By irrelevant context here, we mean that NM conclusions can be drawn in a **local** (intra-slice) fashion, without any need to consider those remaining (external) axioms, in a manner that is **sound** relative to the global theory. (By drawing conclusions here, we mean using the same non-monotonic theory operator $\mathcal{C}$ for the NM formalism as is used to draw conclusions from the global axiom set.)

$$(\mathcal{A} - \mathcal{SA}_i) \ \textit{is irrelevant to} \ \mathcal{SA}_i$$

In this case, we say that the locality is *strong*. The slice axiom set is definitively simpler than the global axiom set. No mutation is required.

Even nicer is when all slices are clean in a particular decomposition.

Examples: totally-prioritized modules in sub-section 5.13.1 (and thus totally-prioritized propositional defaults in sub-section 5.13.2), disjoint sub-languages in section 6.2 (moreover, all slices are clean), sympathetically solitary in section 6.5 (moreover, all slices are clean).

Summarizing Relevant External Context:
More common and still useful, though less ideal, is to expose a weaker kind of locality, in which the relevant context to $\mathcal{SA}_i$ can be *summarized* short of the entire remainder $(\mathcal{A} - \mathcal{SA}_i)$ of the global axiom set. E.g., some but not all, of the axioms in that remainder may be irrelevant to $\mathcal{SA}_i$. Or, a mutation may serve as a kind of set of declarations of external context, analogous to the role of declarations in programming languages with side effects. An important issue then is: in what sense is the declaration mutation simpler than the remainder of the global axiom set. Chapter 5 discusses summarization by mutation much more.

Note that one must think of axiom sets, not just individual axioms, as being involved in relevance. More precisely, by relevance, we mean axioms that affect which non-monotonic, not just monotonic, conclusions can be drawn.

Examples: canonical decomposition in sections 5.4 and 5.5, totally-prioritized modules in sub-section 5.13.1.

Observation 4.9 (Hierarchy)
We can view the conjunctive combination of a set of slice sub-theories as being, in turn, a sub-theory. When those slices are clean, then this sub-theory is itself well-defined as a clean slice: its axiom set is simply the union of those slices' axiom sets. Even when the slicing is not clean, often one can conjunctively decompose a slice into finer slices. Thus:

> *One can slice within slices, and choose grain size hierarchically during conjunctive decomposition.*

This capability is especially important to tune the selectiveness of forward inference.

Examples: canonical decomposition for modules and instances in sub-sections 5.4.4 and 5.4.5, disjoint sub-languages in section 6.2, sympathetically solitary in section 6.5, the running example in chapter 6.

Observation 4.10 (Locality in Specification)
Locality, concurrency, hierarchical scope, and safeties of updating all can help the process of performing specification, especially of large axiom sets when more than person is performing the specification. **Recall:** sub-section 4.3.4. See section 8.2 for more discussion about specification process and knowledge acquisition.

4.5.4 Serial Decomposition and Its Uses

Serial decomposition has the flavor of a cascade: there is a series of phases of adding axioms and drawing conclusions, where the previous stage's conclusions are treated as for-sure. Many NM inference procedures, as well as definitional fixed point constructions can be described in this manner. E.g., in Default Logic and Poole's formalism (see section 8.5), where the fixed point construction draws default conclusions one default at a time, then treats them as non-retractable from then on.

Serial decomposition is more complicated to define than conjunctive decomposition.

Definition 4.11 (Serial Decomposition)
We say that a global axiom set $\mathcal{A}$ is *serially decomposable* into a *sequence* of constituent axiom sets $\langle \mathcal{SA}_i | i = 1, \ldots, n \rangle$ when the global theory $\mathcal{T}$ (i.e., $\mathcal{C}(\mathcal{A})$) is the last-most of the *tier sub-theories* $\langle \mathcal{ST}_i | i = 1, \ldots, n \rangle$ formed from that axiom-set sequence. Each tier sub-theory corresponds to a *tier axiom set* $\mathcal{TSA}_i$:

$$\mathcal{ST}_i \stackrel{\text{def}}{=} \mathcal{C}(\mathcal{TSA}_i)$$

The tiers are defined in an accumulative fashion, as follows. The basic intuition is that consequences are drawn in *phases*, and the results of previous phases are treated as having for-sure, i.e., un-retractable, status. In CLD, we define this in terms of a *classical* axiomatization of of the previous tier sub-theory. We let $\mathcal{TSB}_i$ stand for the axiom set that contains the for-sure assertion of each axiom in some classical axiomatization of $\mathcal{TST}_i$. How this classical axiomatization is broken up into the base axioms of $\mathcal{TSB}_i$ makes no difference. [24] [25]

N.B.: We extend CLD to permit the formula part of a base axiom to be second-order, not just first-order. This is straightforward. When the base axioms in the constituent axiom set sequence have formula parts that are (first- or) second-order, then the resulting circumscriptions and tier sub-theories are second-order classical, and thus matters are well-defined.

The first tier axiom set $\mathcal{TSA}_i$ is simply the first axiom set in the sequence: $\mathcal{SA}_1$. For $i \geq 2$, the i^{th} tier axiom set is the union of the i^{th} constituent axiom set in the sequence with the for-sure assertion of a classical axiomatization of the previous tier sub-theory:

$$\mathcal{TSA}_i \stackrel{\text{def}}{=} \mathcal{TSB}_{i-1} \cup \mathcal{SA}_i$$

Figure 4.7 illustrates the concept of serial decomposition.

In this thesis, for CLD serial decompositions, we will be interested in constituent axiom set sequences in which only the first in the sequence contains base axioms. Each phase circumscription after the first thus takes as its base sentence: the previous phase's circumscription sentence. The

[24] We **assume** another condition on the logical system $\mathcal{S}$ in order to make things well-defined: that the choice of classical axiomatizations is irrelevant; in general, any theory may have many classical axiomatizations. CLD satisfies this condition.

[25] Note that it is possible to generalize all this to many other logical systems, e.g., many modal logics.

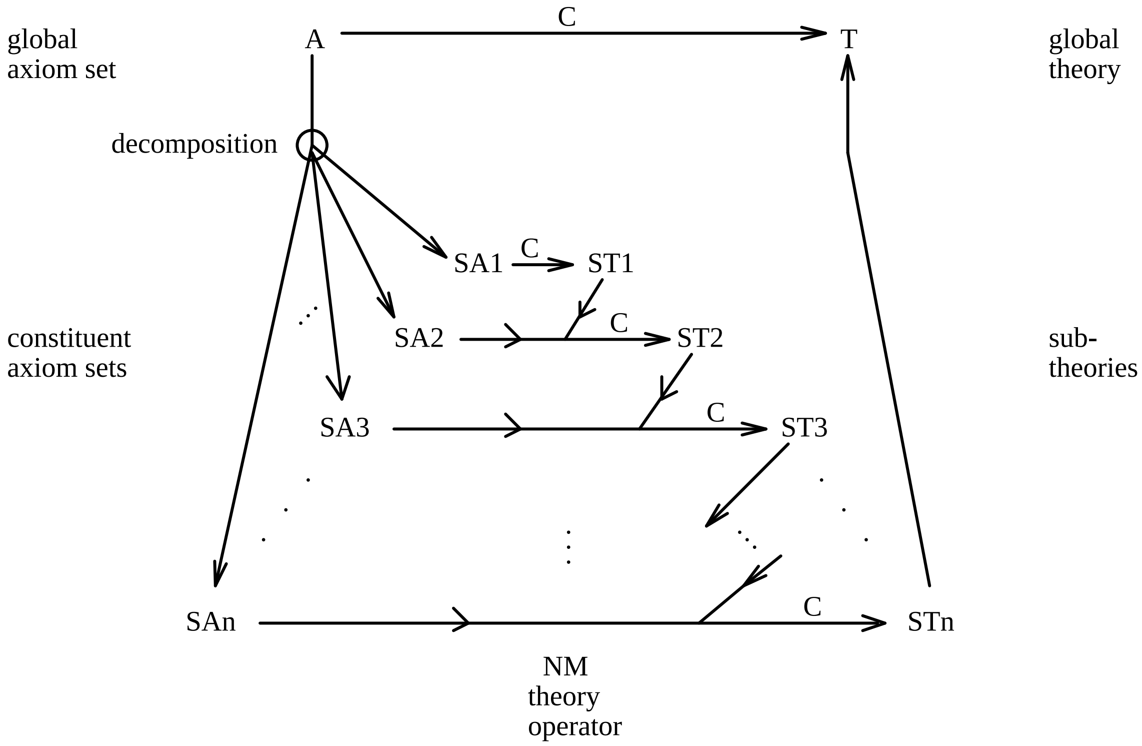


Figure 4.7: Serial Decomposition: a conceptual flow diagram. Similar to conjunctive decomposition, except that each constituent axiom set $\mathcal{SA}_i$ is augmented by (the for-sure assertion of) the previous tier sub-theory $\mathcal{ST}_{i-1}$ before $\mathcal{C}$ is applied to generate the corresponding tier sub-theory $\mathcal{ST}_i$. The final tier in this cascade is then the global theory.

first phase circumscription takes as its base what is specified by the first constituent axiom set: $base\_\mathcal{SA}_1$. Each CLD constituent axiom set $\mathcal{SA}_i$ specifies a circumscriptive policy $policy\_\mathcal{SA}_i$ for the i^{th} phase circumscription.

Definition 4.12 (Series Circumscription)
Let $\langle policy_1, \ldots, policy_n \rangle$ be a sequence (i.e., a tuple) of policies. Then the *series circumscription* (a.k.a. *serial* circumscription) of a base sentence B with respect to this *serial policy* is:

$$\left(\mathop{///}_{i=1,\ldots,n} C(\underset{B}{\bullet}; policy_i) \right) \stackrel{\text{def}}{\equiv} C(\ldots C(C(B; policy_1); policy_2) \ldots; policy_n)$$

More generally, we define the series circumscription corresponding to a sequencing of a tuple of policies. We say that σ is a *sequencing* (a.k.a. *sequence*) of the n indices in the index tuple N when σ is a bijective map from $\{1, \ldots, n\}$ into the set N. Thus the sequence $\langle \sigma(1), \sigma(2), \ldots, \sigma(n) \rangle$ is a permutation of the tuple N: $\sigma(1)$ is the first index in the sequence, $\sigma(2)$ is the second, and so on.

$$\mathop{///}_{\substack{i \in N \\ via \\ \sigma}} C(\underset{B}{\bullet}; policy_i) \stackrel{\text{def}}{\equiv} C(\ldots C(C(B; policy_{\sigma(1)}); policy_{\sigma(2)}) \ldots; policy_{\sigma(n)})$$

Then, for a CLD serial decomposition: in terms of the circumscriptions involved, we have

$$\mathcal{PDC}(\mathcal{A}) \equiv \mathop{///}_{i=1,\ldots,n} \mathcal{PDC}(\underset{base\_\mathcal{SA}_1}{\bullet}; policy\_\mathcal{SA}_i)$$

Again, when the corresponding axiom sets are understood, we will also speak of a *circumscription* being serially decomposable into phase circumscriptions, e.g., for $n = 2$:

$$PDC(B; D; R; Z) \equiv PDC(PDC(SB1; SD1; SR1; Z); SD2; SR2; Z)$$

Note that two basic properties related to serial decomposition are: cumulativity (see sub-section 4.8.1); and *idempotence*.

Fact 4.13 (Series Stays Second-Order)
Note that for prioritized-default series circumscriptions, the base for the second and later phases is, in general, second-order rather than first-order. However, the series circumscription is still second-order.

Fact 4.14 (Idempotence)
Any circumscription is idempotent:

$$C(B; H; Z) \equiv C(C(B; H; Z); H; Z)$$

Idempotence implies that multiple appearances of a policy in a serial policy are equivalent just to the first appearance alone; i.e., one can equivalently delete the further appearances from the policy sequence.

Proof : When the policy is applied for the second, or later, time, the models of the base sentence (at that point in the series) are already minimal with respect to that policy. □

Observation 4.15 (Uses of Serial Decomposition)
The uses offered by serial decomposition are similar to those offered by conjunctive decomposition. The NM inference can be performed locally, within each phase; and hierarchically, since phases may themselves be further decomposed into a finer grain. Note that:

> *Conjunctive and serial decomposition may be mixed: slices may be decomposed into phases, and vice versa.*

As we will show later, often the sequence of the constituent axiom sets in a serial decomposition can, equivalently, be permuted in many ways.

> *Alternative permutations in serial decomposition can expose opportunities for alternative sequencing and concurrency in inference.*

If the first k in some sequencing are left unaffected by an update, then we know that all of the consequences in the k^{th} tier are safe under that update. Exhaustive inference on a tier generates only a selective part of the global theory. Inference can thus be performed selectively, by choosing an appropriate sequencing as well as grain size. Finally:

> *Serial decompositions can help to understand relationships between different NM logical formalisms.*

(See remarks in first paragraph of this sub-section.)

Definition 4.16 (Clean Phase)
Similarly to the idea of a clean slice, we can also define a *clean* phase as one in which the constituent axiom set is a subset of the global. Matters are especially nice when a clean phase comes first in the sequencing, or, more generally, when the first k in a sequencing are clean.

Examples: totally-prioritized modules in sub-section 5.13.1, totally-prioritized propositional defaults in sub-section 5.13.2.

Perspective: Reformulation:
Decomposition can be viewed as a kind of reformulation: a global axiom set / theory is reformulated into slices or phases / tiers.

4.6 Strategy for Scale: Recap

In this section, we recap the analysis and divide-and-conquer strategy laid out earlier in this chapter.

Recall: sections 1.3 and 1.4.

The problem of scale in expressively rich NMR arises in specification, in inference, and in updating (belief revision). We analyzed the difficulty of this problem: as due to logical globality and non-modularity.

We reformulated these problems of scale as:

- how to "beat globality": i.e., to achieve a degree of locality of context-dependence, i.e., a degree of modularity, in specification and inference; and
- how to achieve a zone of safety in updating.

These two problems are closely related. In a sense, they are duals. For a given axiom (sub)set:

- Safety = which conclusions are preserved for a given update.
  Safety = partial monotonicity of updating.
- Relevant context = complement of irrelevant context.
  Irrelevant context = which (other axioms considered as "internal") updates preserve a given conclusion (or set of conclusions, e.g., the whole non-monotonic theory corresponding to the given axiom (sub)set).

Thus the study of modularity/ locality/ (ir)relevance and the study of safety of updating are closely related.

Terminology: Henceforth, we will refer to them together: as study of **locality**.

Our aim, therefore, is to understand better the interaction between parts of axiom sets and theories. Our first point of overall strategy is, accordingly, to organize the global axiom set and the sanctioned conclusions into locales, on the basis of global information, where one can relatively quickly and easily summarize the relevant external context. One would like this organization into locales to be relatively stable during updating. In addition, we would like them to tell us something about opportunities for concurrency and constraints on sequencing during the process of inference.

We care especially about the task of belief revision in updating. This is directly important in the kinds of (potential) applications that motivate us most: learning agents, in the realm of empiricism and action. More broadly, belief revision makes visible much of the underlying problems in the tasks of inference and specification. We observed that one can view external context in inference and in specification as a kind of internal updating.

Our second point of overall strategy is to attack the problem of scale at a logical level: to consider the entire set of sanctioned conclusions. Rather than, say, at "the domain level" or "the procedural level": trying to develop domain-dependent heuristics for control of inference (and updating) procedures. Our strategy is thus quite domain-independent, but does not directly bear on understanding the effect of resource constraints on reasoning.

We would like for special cases of modularity and safety to be easily recognizable by future automated reasoners and/or their human designers. We will, therefore, show how such cases can often be described in terms of relatively simple **syntactic** conditions on the axiom sets and conclusions involved. In doing so, we will treat precedence information as part of an extended notion of syntax.

As a step towards automating the use of our results about decomposition, monotonicity of updating, and alternative sequencing in inference, we will define a "genealogical" truth maintenance scheme, in which conclusions "descend" to the current global NM theory in two ways: 1) from previous global NM theories via safeties; and 2) from current NM sub-theories via decompositions. This scheme provides a tool to do the "book-keeping" when designing procedures for NMR. We will, further, build a "genealogical" NMR architecture around the genealogical truth maintenance scheme.

In more detail: We will

- introduce ideas of structure in axiom sets and in theories
- including, the ideas of hierarchical conjunctive and serial decomposition as a basis for a strong notion of a part
- on the basis of syntactic relationships so as to be easily recognizable
- including, precedence as a kind of syntactic relationship
- relate structure among axioms to structure among conclusions.
- find special cases with lots of irrelevance
- summarize relevant context in a manner analogous to declarations in programming languages with side effects
- develop theorems on decomposition
- exploit structure to get safety: use the "differential" of decomposition before and after updating to obtain theorems on partial monotonicities of updating
- develop other theorems on monotonicities of updating
- also use decomposition to obtain results on concurrency and sequence constraints on the process of inference
- develop a "genealogical" truth maintenance scheme and NMR architecture to support automated processes of inference and belief revision

4.7 Concepts of Behavior Under Updating

In this section, we introduce some concepts of behavior of circumscriptive theories under updating. These also apply to *any non-monotonic formalism.*

N.B. In the remainder of this thesis, we will refer interchangeably (unless explicitly distinguished in context) to a circumscription and to its corresponding circumscriptive theory. For the most part, we will be speaking of CLD and prioritized default circumscription. Thus we will also refer interchangeably to updating a CLD axiom set and to updating a prioritized default circumscription.

Assumption: Axiom Sets Are Finite:

Unless explicitly mentioned otherwise, in the remainder of this thesis we assume all CLD axiom sets are finite, and that all prioritized default circumscriptions correspond to finite CLD axiom sets.

Definition 4.17 (Monotonic and Redundant, Current and Forever)

We define an update to be *(globally) monotonic* if the new circumscription implies the old, i.e., if the circumscriptive theory entails (includes) the old. We define the update to be *redundant* if the new theory is equivalent to the old. Redundancy is a special case of monotonicity. We say that an update is *strictly* monotonic when it is monotonic but not redundant. We say that an update is *non-monotonic* when it is not monotonic. More formally: Let $\mathcal{A}$ and $\mathcal{U}$ each be a CLD axiom set. $\mathcal{U}$ is a *(currently) redundant* update to $\mathcal{A}$ when the conclusions are the same after the update:

$$\mathcal{A} \Vdash\!\!\!\approx \mathcal{U} \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad \mathcal{C}(\mathcal{A}\&\mathcal{U}) \equiv \mathcal{C}(\mathcal{A})$$

$\mathcal{U}$ is a *(currently) monotonic* update to $\mathcal{A}$ when the update does not require the retraction of any previous conclusions:

$$\mathcal{C}(\mathcal{A}\&\mathcal{U}) \models \mathcal{C}(\mathcal{A})$$

We say that $\mathcal{U}$ is a *forever redundant* update to $\mathcal{A}$ when after *any* further updating, i.e., by some $\mathcal{W}$, a finite set of CLD axioms, the conclusions are the same with and without the update $\mathcal{U}$:

$$\mathcal{A} \Vdash\!\!= \mathcal{U} \quad \stackrel{\text{def}}{\Longleftrightarrow} \quad \forall \mathcal{U}.\, \mathcal{C}(\mathcal{A}\&\mathcal{U}\&\mathcal{W}) \equiv \mathcal{C}(\mathcal{A}\&\mathcal{W})$$

Similarly, we say that $\mathcal{U}$ is a *forever monotonic* update to $\mathcal{A}$ when after any further updating, i.e., by some $\mathcal{W}$, a finite set of CLD axioms, the conclusions are at least as strong with the update $\mathcal{U}$ as without it:

$$\forall \mathcal{W}.\, \mathcal{C}(\mathcal{A}\&\mathcal{U}\&\mathcal{W}) \models \mathcal{C}(\mathcal{A}\&\mathcal{W})$$

Note that forever monotonicity (or redundancy) means that one can "wait until later" to "process" the update, and still have monotonicity (or redundancy). Forever redundancy can be viewed as a kind of monotonic *entailment*: if an axiom is forever-redundant, it is in effect always present in the axiom set. This has the feel of non-monotonic formalisms that put default axioms into the language, e.g., modal approaches such as Autoepistemic Logic [Moore, 1985] and Delgrande's conditional logic [1987b].

Results on Forever Monotonicity and Redundancy: Several of our results can be viewed in terms of forever monotonicity and redundancy. For Example, Theorem 2.58 implies that **prioritization updates are forever monotonic**. Also, our results in section 3.5 imply forever redundancies of many fixture updates. Finally, our results on "strong sympathy" in section 6.6 are directly about guarantees of forever monotonicity and forever redundancy.

The spirit of the above definitions goes beyond updating. What should be an appropriate idea of equivalence between non-monotonic *axiom sets* within a given formalism, as opposed to between non-monotonic *theories*, is somewhat problematic. Clearly, it is unsatisfactory to simply say, for example, that a CLD axiom set containing defaults is "equivalent" to a set of base axioms that happen to axiomatize its CLD theory, and leave it at that. This loses most of the point of employing non-monotonic, as opposed to monotonic, logic. Accordingly, we define a stronger idea of equivalence than just *current* equivalence.

Definition 4.18 (Forever Equivalence)
We say that two CLD axiom sets $\mathcal{A}_1$ and $\mathcal{A}_2$ are *forever equivalent* when they result in the same theory after *any* update $\mathcal{W}$:

$$\forall \mathcal{W}.\ \mathcal{C}(\mathcal{A}_1 \& \mathcal{W}) \ \equiv \ \mathcal{C}(\mathcal{A}_2 \& \mathcal{W})$$

Forever redundancy is a special case of forever equivalence: the axiom sets differ only by the update.

More generally, it is interesting to consider **forever equivalence under restricted classes of updates, and on restricted sub-languages** of the base language. Many of our results in chapter 3, especially in sections 3.2 and 3.5, imply such forever equivalences; next we mention a few of them. (Also see our results about forever redundancy in section 6.6.)

Results on Forever Equivalencies: For example, a direct consequence of Theorem 3.8 is that any CLD axiom set is forever equivalent to its corresponding definitional **abnormality** theory's CLD axiom set: on the abnormality-free sub-language.

And Theorem 3.8 also implies directly that any CLD axiom set is forever equivalent, on the abnormality-free sub-language, to its corresponding *adaptive* abnormality theory's CLD axiom set, when the updates are restricted not to mention any of the abnormality predicates. More generally, the proof of Theorem 3.8 implies that, in any CLD axiom set, it is forever equivalent, on the abi-free sub-language, to:

1) replace the default axiom

$$(di) :> \ Di[Z, x]$$

with the pair of axioms

$$\bullet\!> \ \forall x.\ \neg abi(x) \equiv Di[Z, x]$$
$$(di) :> \ \neg abi(x)$$

, where abi is new predicate symbol;
2) or, under abi-free updates, to replace it with the pair of axioms

$\bullet\!> \ \forall x.\, \neg abi(x) \supset Di[Z, x]$

$(di) :> \ \neg abi(x)$

Or, as another example, the proof of Corollary 3.46 implies directly that any CLD axiom set $\mathcal{A}$ containing a fixture axiom with formula part $F[Z, x]$ is forever equivalent to the CLD axiom set formed from $\mathcal{A}$ by replacing that fixture axiom with the pair of axioms

$\bullet\!> \ \forall x.\, par(x) = F[Z, x]$

$\Box\!> \ par(x)$

, where par is a new predicate symbol, when the updates are restricted not to mention par. Furthermore, Lemma 3.42 implies directly that $\mathcal{A}$ is also forever equivalent to the CLD axiom set formed from $\mathcal{A}$ by replacing that fixture axiom with the pair of axioms:

$:> \ F[Z, x]$

$:> \ \neg F[Z, x]$

Here the defaults are unlabelled. This implicitly specifies that no updates will affect their prioritization.

4.8 Tastes of Trouble in Updating; Conditional Entailment Perspective

The analogues of many basic properties that one takes for granted in classical monotonic logic, fail to hold in prioritized default circumscription, and in many other formalisms for NM / default reasoning. This indicates that characterizing monotonicity of updating even at the level of the logic, computational challenges aside, is a tricky enterprise. One has to be careful about bringing intuitions from the monotonic case.

In this section, we discuss some of these trampled properties, and related them to conditional entailment cf. Kraus *et al* [1990]. This helps to motivate our focus, in this thesis, on study at the level of the logic.

4.8.1 Non-"Well-Foundedness"; Cumulativity

In section 3.4, we discussed the fact that circumscription, even parallel predicate circumscription, can be badly-behaved in a very basic way: nasty unsatisfiability.

Less obviously, violation of "well-foundedness" (see Definition 3.16), even when satisfiability is preserved, can lead to intuitively strange behavior.[26] As we remarked in section 3.4, the possibility of non-"well-foundedness" is a difficulty for many NM formalisms definable in terms of preferred models, not just for circumscription.

[26] [Lifschitz, 1986] shows, by example, that circumscription can be satisfiable yet not "well-founded".

One basic property that can be violated in case of non-"well-foundedness" is *cumulativity* [Makinson, 1989]. Cumulativity is the property that updating with a previous conclusion as a for-sure (base) axiom is (currently) redundant. I.e., in circumscription:

$$[C(B; pol) \models E] \implies [C(B \wedge E; pol) \equiv C(B; pol)]$$

Conditional Entailment cf. Kraus, Lehmann, and Magidor

Kraus, Lehmann, and Magidor ([Kraus *et al.*, 1990] [Lehmann and Magidor, 1988] [Lehmann, 1989]) [27] study properties of the space of non-monotonic entailment relations. That is, they consider implications among various different potential properties of $\approx$, abstracted away from any particular NM formalism. In our terms, they primarily consider only updates with new base axioms, i.e., with new for-sure premises.

When (general) circumscription is "well-founded", it is what Kraus, Lehmann and Magidor [1990] call a **preferential logic**. They show that circumscription in that case therefore obeys cumulativity and several other basic properties. [28]

In section 3.4, we discussed expressive conditions ensuring "well-foundedness". The good news is that, as we observed there, these conditions are often met in practical AI Knowledge Representation.

We **observe** that Theorem 3.24 implies that quasi-propositional prioritized default / predicate circumscription, for example, is a preferential logic in Kraus *et al*'s sense, even when the prioritization is non-layered.

4.8.2 Failure of Weakening For Defaults

Kraus, Lehmann and Magidor did not consider, however, the question of updating with other than base axioms, e.g., with new defaults.

Next, we show that updating with defaults behaves in a way that runs counter to intuitions drawn from monotonic (classical) logic, and makes for difficulty in belief revision.

Observation 4.19 (Failure of Weakening for Defaults)
We take for granted in classical logic that D entails any weakening $D \vee E$. (Where D and E are elementary formulas.) However, the analogue for defaults does not hold. Even for the propositional case without priorities (i.e., where all defaults are in parallel), in no sense does $:> \; D$ entail[29] $:> \; D \vee E$. In general, if the former is present in a CLD axiom set $\mathcal{A}$, updating with the latter may be redundant, strictly monotonic, or non-monotonic. Indeed, for any logically independent D and E, we can construct a simple example of each kind.

Proof : **I)** Redundant: Let $\mathcal{A}$ consist of just $:> \; D$. Then before the update, the CLD theory is equivalent to D. After the update, it still is.

[27] See also closely related work by Makinson [Makinson, 1989] on cumulativity and related properties.

[28] Their "stoppered"-ness condition is, essentially, "well-foundedness".

[29] monotonically, in the sense we discussed in Definition 4.17

II) Strictly monotonic: Let $\mathcal{A}$ consist of $\bullet\!>\ \neg D$ and $:>\ D$. Then before the update, the theory is equivalent to $\neg D$. After the update, the theory is equivalent to $\neg D \wedge E$.

III) Non-monotonic: Let $\mathcal{A}$ consist of $\bullet\!>\ \neg D$, $:>\ \neg E$, and $:>\ D$. Then before the update, the theory is equivalent to $\neg D \wedge \neg E$. After the update, the theory is equivalent to $\neg D$. □

Special Case: Conjunctive Default Formula:
In particular, a default whose formula is a conjunction:

$:>\ D1 \wedge D2$

does *not* entail the default version of either its conjuncts:

$:>\ D1$

$:>\ D2$

Interestingly, though, we show later (Corollary 6.30) that $:>\ D1$ and $:>\ D2$ together *do* entail $:>\ D1 \wedge D2$.

Same For Other NM Formalisms:
Note that all of the examples in the proof of Observation 4.19 are propositional (and, therefore, by Theorem 3.24, "well-founded") and without priorities or fixtures: the case most common to various formalisms for default reasoning. Their translation into many other NM formalisms, e.g. Default Logic, shows that those formalisms have the same qualitative behavior.

Note, furthermore, that these examples can easily be expressed, via abnormalities, in predicate circumscription instead of default circumscription.

The next two examples elaborate, and give some intuition as to how a logically-weaker-formula default can still yield new conclusions, or else how it can require retraction. Imagine the "before" axiom set as having been accumulated by a series of updates.

Example 4.20 (Bullies)

$:>\ bully(x)\ \supset\ big(x) \wedge strong(x)$

$\bullet\!>\ bully(Napoleon)$

$\bullet\!>\ \neg big(Napoleon)$

when updated by

$:>\ bully(x)\ \supset\ strong(x)$

results in the new conclusion that $strong(Napoleon)$.

In terms of this example: with a default formed by conjoining two conjuncts $D1$ and $D2$, it may be that one conjunct is defeated but not the other.

Example 4.21 (Party-Going)

$:>\ party\_bound(Don)$

$:>\ party\_bound(Eric)$

$\bullet\!>\ \neg party\_bound(Don)$

when updated by

$:>\ party\_bound(Don) \vee \neg party\_bound(Eric)$

results in the retraction of the conclusion that $party\_bound(Eric)$.

4.8.3 Failure of Rational Monotony

Lehmann and Magidor [1988] propose a condition that they call *rational monotony*, as a restriction on preferential logics (recall page 139), and study NM logical systems that satisfy it.

In circumscriptive terms, "rational"[30] monotony is defined as:

$$\forall B, E, U. \;\; [C(B; pol) \models E] \;\wedge\; [C(B \wedge U; pol) \not\models E] \;\;\supset\;\; [C(B; pol) \models \neg U]$$

which is equivalent to:

$$\forall B, E, U. \;\; [C(B; pol) \models E] \;\wedge\; [C(B; pol) \not\models \neg U] \;\;\supset\;\; [C(B \wedge U; pol) \models E]$$

(B, E, and U are sentences.)

Observation 4.22 ("Rational" Equals Consistent)
"Rational" monotony is thus equivalent to the monotonicity of any for-sure (base) update that is consistent with the previous theory:

$$\forall B, U. \;\; C(B; pol) \not\models \neg U] \;\;\supset$$
$$\{\forall E. \, [C(B; pol) \models E] \;\supset\; [C(B \wedge U; pol) \models E]\}$$

If "rational" monotony holds, then there is a conceptually simple test for whether a base update U is globally monotonic: if it is consistent with the previous theory, then yes; else no. (If it is inconsistent with the previous theory, then its negation $\neg U$ must be retracted, so the update is non-monotonic.)

However, matters are more complicated.

Observation 4.23 (Rational Monotony Fails)
Rational monotony fails for even the parallel propositional predicate case of circumscription (which by Theorem 3.24 is "well-founded').

Proof Overview: In the Appendix: We give a counterexample, which involves only three defaults and no fixing.[31] □

Same For Other NM Formalisms:
Straightforward translation of the counterexample into many other NM formalisms, e.g. the skeptical version of Default Logic using Observation 8.1 in section 8.5, shows that those formalisms also fail to obey rational monotony.

Lehmann and Magidor show that rational monotony is equivalent to the condition that the models be *ranked* i.e., layered, under the preference pre-order. (Not to be confused with the layered-ness of the prioritization partial order, which does not ensure it, as the counterexample shows.)

However, ours is the first counterexample for circumscription, that we are aware of, in the NMR literature, showing its violation of rational monotony.

[30] We use scare quotes since we do not find the condition reasonable to require of all rational agents.

[31] This example was worked out jointly with Hector Geffner.

An interesting open question is to refine the intuitions that justify "rational" monotony. However, we do not find it at all reasonable to *require* it for rationality.

Chapter 5

Prioritization As Structure

In this chapter, we develop some general decomposition theorems using priority. Our investigations illuminate the effects, and mathematical complexity, of non-layered prioritization. We introduce the concept of "guarding" to represent the structure of potential conflicting interaction. Guards play a role analogous to that of declarations in programming languages with side effects. We then apply the decompositions to obtain theorems about safeties of updating based on priority.

Finally, we develop some results about the strongly locality for special cases where prioritization is totally ordered. These results include: decompositions; safeties of updating; and a new forward-inference algorithm.

Guide to Reader: Section 5.1 provides a summary / overview of the whole chapter. Recall sections 4.5 and 4.6 before continuing on.

5.1 Introduction and Summary

In this chapter, we study prioritization as a basis for structure in NM *theories*, and relate it to prioritization as a basis for structure in axiom sets. This corresponds to the strategy of attack on scale that we discussed in sections 4.3 through 4.6. Recall especially Figure 4.5 (section 4.4) and the Observations in section 4.5.

We show (sections 5.4 and 5.5) how to decompose ANY prioritized circumscription, e.g., corresponding to a CLD axiom set, along its prioritization, either conjunctively or serially, into one slice or phase per starting pre-order, e.g., per default. This method of decomposition applies also to the parallel case, which is suggestive for other NM formalisms that lack the ability to express prioritization. These results generalize those of Lifschitz, who studied the case of predicate circumscriptions with layered priority.

The NON-LAYERED case of priority turns out to be MUCH MORE COMPLICATED to decompose. Yet, as we observed in chapters 1 and 2, the non-layered case is **much more IMPORTANT in applications** of default reasoning.

These decompositions employ a *canonical protection tactic* that, for each slice or phase policy, e.g., for each default, summarizes its relevant external context, via a concept of guarding (section 5.3). Guarding is achieved by introducing new fixtures and new base axioms. These fixtures, in effect, impose monotonicity constraints; remember that fixed predicates and formulas are immune to the non-monotonic effect of circumscription (Theorems 3.38 and 3.39). The tactic is: in the slice or phase where a default i is maximized, to protect each external default that is higher- or incomparable- priority than i. Each higher-priority default is protected by fixing its default formulas. Each incomparable-priority default is protected by guarding its default formula. The lower-priority defaults are *irrelevant* (in the sense of Observation 4.8): they do not need to be protected. There is a further subtlety, however: the prioritization among the incomparable-priority defaults needs to be represented in the guarding.

We show that this protection tactic is robust in two senses. Firstly, it can be applied in both conjunctive and serial styles of decomposition. Secondly, even in serial decomposition, it can often be applied without regard to sequentiality.

We show (subsection 5.4.4), further, that this canonical decomposition along prioritization can be applied hierarchically, i.e., recursively, when the global prioritization is built up by composition of prioritization (subsection 2.7.3), e.g., from modules (Definition 2.62). One can slice within slices, for example, corresponding to sub-modules within modules. In such a case, one can thus choose the desired grain size of decomposition.

Indeed, we show (sub-section 5.4.5) that the decomposition can, often, continue right down to the grain size of one *closed* default *instance* per slice or phase. Domain closure plus uniqueness of names suffice. We exploit this instance-grain decomposition and perspective repeatedly in our special-case strong-locality results later (sections 6.3 and 6.5).

Our decomposition results have a number of pay-offs. Firstly, they show how to *expressively reduce prioritization*: to eliminate it completely, in a sense; or, more generally, to simplify it (in case of hierarchical decomposition). (See especially section 5.7 for discussion.)

Secondly (see especially section 5.6), these decomposition results illuminate the nature of defaults' interaction and context-dependence. We show that lower-priority defaults are, in the sense of Observation 4.8, irrelevant to inference from higher-priority defaults. The details are especially non-trivial, however, when the prioritization is non-layered. We discover (section 5.10) an interesting analogy between the constituent CLD axiom sets in these decompositions, and sub-programs in programming languages with side-effects. In such programming languages, one summarizes relevant external context with declarations about non-locals read and written. In our NM decompositions, the role of a non-local variable in a programming language is played by the truth value of a non-local default's formula. (By non-local default, here, we mean a default axiom that is present in the global axiom set but that is not present as a default axiom in the given constituent axiom set under consideration.)

Thirdly, our decomposition theorems give mathematical insight and are useful for proving properties of prioritized circumscriptions, e.g., for hand-computing (see section 5.9, in particular) the conclusions entailed by a particular circumscription. They provide a foundation for special case results: including those in section 5.13 and chapter 6.

Fourthly (section 5.8), we show how to define a canonical concept of derivability from a subset of the global defaults; and more generally, derivability from a subset of the starting pre-orders. The "obvious" simplistic approach is to say a conclusion is derived from a subset of defaults if those defaults (and their prioritization) plus the base entail the conclusion. However, this is unsound, in general, because it may be that some of the rest of the global set of defaults conflict with some of the defaults in this subset. We use, instead, the conjunctive combination of the canonical slices that correspond to each default in the subset. This ensures soundness, as well as completeness in the sense discussed in Observation 4.3.

We use this notion of derivability, together with the decomposition results, to show how to enable locality, selectiveness and concurrency in inference, in the senses we discussed in section 4.5 (Observations 4.4 and 4.5): see section 5.9 and chapter 7 (especially section 7.5).

Fifthly (section 5.12), our decomposition results yield safeties of updating. We use the notion of canonical derivability to show a general result about the safety of higher-priority conclusions, i.e., the safety of all previous conclusions derived solely from the subset of the defaults that are higher-priority than some new update defaults. Again, the "obvious" simplistic approach fails: non-layered prioritization makes things much trickier than in the layered case from which you might draw your intuitions. We use canonical derivability to define just what are those higher-priority conclusions.

A difficulty with the general case of our canonical decomposition results, however, is that the slices or phases are not always clearly simpler than the global circumscription. By the criterion of fewer defaults (and simpler prioritization), they always are. But there are other possible criteria: other dimensions of the expressive complexity of the slices or phases include the base and fixture axioms.

We analyze (section 5.11) this expressive complexity issue. For example, each slice (or phase) contains at least as many base axioms as the global. And the additional base axioms in each slice or phase may involve an increase in expressive complexity relative to the global base, in a way that goes beyond just an increase in size. In particular: even when the global base and default formulas are universal (e.g., clausal), the base axioms introduced during guard reduction may involve existential quantifiers, when the guards being reduced themselves have prioritization. Furthermore, if one views the introduced fixtures as pairs of polar defaults (Lemma 3.42), then each slice or phase contains at least as many defaults as the global. Intuitively, it seems to us that a fixture is usually simpler than a default. However, it is currently unclear how fixtures affect the computational complexity of inference, both in current inference procedures for circumscription, and intrinsically.

We show (section 5.11) that the decompositions can be computed in time that is polynomial (worst-case cubic) in the size of the global axiom set. One implication of this result is that the expressive reducibility they furnish, e.g., of prioritization, is non-trivial in a particular sense.

Sixthly, however, in many special cases, the decompositions via priority result in some slices or phases that *are* clearly simpler than the global.

We show (sub-section 5.13.1) that one such special case is when there is total (Definition 2.53) prioritization between modules. Then there is always a clean slice or phase (Definitions 4.7 and 4.16). And the base never becomes more expressively complex in the above-discussed senses.

A yet more special case is when all defaults are propositional, and there is total prioritization among them. We show (sub-section 5.13.2) that in this case a particularly simple form of serial decomposition obtains, in which there is no need to protect (guard). Furthermore, each phase is just equivalent to a simple consistency check for that phase's default. This yields a new forward-inference algorithm, and is suggestive for more algorithms and for comparison to other NM formalisms defined in terms of serial consistency. This result is less obvious than it first appears. We show that there is a subtlety: no global fixtures may be allowed.

In addition, we show (subsection 5.13.3) that one can view base axioms as a special case of prioritized default axioms: as defaults at highest-priority. Updating with base axioms is thus, essentially, a special case of updating with prioritized default axioms. This supports our effort to formalize (prioritized) defaults as axioms in circumscription. And it reinforces our interest in the problem of updating with new defaults.

In chapters 6 and 7, we exploit our decomposition results further and discuss more of their implications there and in chapter 8.

Guide to Reader: In sections 5.2 through 5.5, we lay out the mathematical details of the decompositions along prioritization, and postpone most discussion until sections 5.6 through 5.11. We defer most discussion of implications for design of reasoning procedures until chapter 7.

5.2 Previous Results

There are three major previous results about decomposition in circumscription on the basis of prioritization, all due to Lifschitz. All were restricted to the *layered* case of prioritization, and to predicate circumscriptions. He did not use the concept of (de)composition of prioritization, nor the concepts of (de)composition of theories (they are new with us).

Theorem 5.1 (Conjunctive, Predicates by Layers (Lifschitz))
Lifschitz showed ([1985]) that:
Any layered-prioritized predicate circumscription can be conjunctively decomposed into one slice per layer, where, in each slice, the higher-priority-layers' predicates are fixed. More precisely: Let N be the global index tuple for the minimized predicates P^N. Let $N1, \ldots, Nk$ be the layers according to the prioritization R, where $N1$ is the highest-priority layer, and Nk is the lowest, so that each P^{Ni} is the tuple of predicates in layer i. (Note that, in our terms, each layer has empty / parallel internal prioritization.) Let NCi stand for $N1 \cup \ldots \cup Ni$, i.e., the union of layer i with all layers that are higher-priority than i. Then

$$PPC(B; P^N; R; fix\ W; Z) \equiv \bigwedge_{i=1}^{k} PPC(B; P^{Ni}; \emptyset; fix\ P^{NCi-1}, W; Z)$$

(where $NC0$ is the empty set.)

Theorem 5.2 (Serial, Predicates by Layers, Descending (Lifschitz))
Lifschitz also showed (unpublished, generalizing Proposition 8 of [Lifschitz, 1984]) that:
Any layered-prioritized predicate circumscription can be serially decomposed, in descending direction of priority, into one phase per layer, where, in each phase, the higher-priority-layers' predicates are fixed. More precisely: Let the notation be as in Theorem 5.1. Then

$$PPC(B; P^N; R; fix\ W; Z) \;\equiv\; \underset{\substack{i=1,\ldots,k \\ via \\ \langle 1,\ldots,k\rangle}}{///} \; PPC(\underset{B}{\bullet}\,; P^{Ni}; \emptyset; fix\ P^{NCi-1}, W; Z)$$

Theorem 5.3 (Conjunctive, Predicates, Parallel, Positivity (Lifschitz))
Finally, Lifschitz showed ([1987b] Proposition 5) that:
If, in a parallel predicate circumscription, all appearances of the minimized predicates in the base are positive[1], then the circumscription can be conjunctively decomposed into one slice per minimized predicate, in each of which just that predicate is minimized and all the other globally-minimized predicates are fixed. More precisely: In a parallel predicate circumscription $PPC(B; P^N; \emptyset; fix\ W; Z)$, suppose that the base B is positive in the minimized predicates P. (We observe that any parallel adaptive-form abnormality theory fulfills this condition, for example.) Then

$$PPC(B; P^N; \emptyset; fix\ W; Z) \;\equiv\; \bigwedge_{i\in N} PPC(B; Pi; \emptyset; fix\ P^{N-i}, W; Z)$$

where $N - i$ stands for all of N other than i.

Discussion: Interestingly, in all three of these results, the interaction between the different minimized predicates is mediated by introducing fixtures. However, this fixing in the first two is asymmetric: higher-priority predicates are fixed when lower-priority predicates are minimized, but not vice versa. The fixing in Theorem 5.3, by contrast, is symmetric: when each predicate Pi is minimized, each other predicate Pj is fixed, and vice versa.

Also, interestingly, the role of fixture was precisely the same in the phases of the layered serial decomposition as in slices of the layered conjunctive decomposition.

In this chapter, we will show how these characteristics generalize to any prioritized circumscription, not just layered predicate circumscriptions.

These three results, and the role of fixing as a kind of protection condition, were part of what inspired us to try to develop the idea of decomposition on the basis of priority, and more general results about it.

Challenge: Non-Layered Prioritization:
The applicability of these previous results is limited, however. As we observed in chapters 1 and

[1] If you are unfamiliar with the terminology of "positive" in logic, see page 225.

2, *non*-layered prioritization is necessary for most large-scale applications of default reasoning. Unfortunately, it turns out that decomposition on the basis of priority becomes MUCH more complicated when the prioritization is non-layered.

5.3 Guarding

In this section, we develop the idea of guards in circumscription, which we will use to formalize the canonical protection tactic.

As a starting point, we observe that the equivalence in Theorem 5.3 is not true, in general, when the positivity condition is relaxed. It is instructive to consider why affairs become less simple to describe in terms of fixtures.

Example 5.4 (Parallel Without Positivity)
Let $B \stackrel{\text{def}}{\equiv}$

$$\begin{aligned} &\forall x.\ Q(x) \supset (P1(x) \equiv P2(x)) \\ &\forall x. \neg Q(x) \supset (P1(x) \vee P2(x)) \end{aligned}$$

Then $PPC(B; P1, P2; \emptyset; \mathit{fix}\ Q; P1, P2) \quad \equiv$

$$\begin{aligned} &\forall x.\ Q(x) \supset (\neg P1(x) \wedge \neg P2(x)) \\ &\forall x. \neg Q(x) \supset (P1(x) \equiv \neg P2(x)) \end{aligned}$$

But this is not equivalent to the decomposition in Theorem 5.3. Indeed, it is not equivalent to the conjunctive decomposition corresponding to any of the four combinations of fixing versus varying $P2$ when $P1$ is minimized, and vice versa. Intuitively speaking, $P1$ and $P2$ need to co-operate in becoming smaller when they (more precisely, their minimizations) do not conflict, as when Q is true, but they need to be protected against each other when they do conflict, as when Q is false. What is needed is a more general sort of protection condition than simply fixing the other minimized predicate. The following equivalence gives some insight:

Observation 5.5 (Parallel Interaction; Need for Guards)

$$\begin{aligned} PPC(B; P1, P2; \emptyset; \mathit{fix}\ Q; P1, P2, Q) \quad \equiv \\ & B[P1, P2, Q] \\ & \wedge\ \neg\exists P1', P2'.\ B[P1', P2', Q] \wedge (P1' < P1) \wedge (P2' \leq P2) \\ & \wedge\ \neg\exists P1', P2'.\ B[P1', P2', Q] \wedge (P2' < P2) \wedge (P1' \leq P1) \end{aligned}$$

The last conjunct in each of the existentials above represents, intuitively, a protection condition. E.g., when one predicate, e.g., $P1$, is minimized, it cannot be at the expense of making the other, e.g., $P2$, become larger; however, "co-operating" with the other's minimization is OK. We call such a protection condition a *guard*.

In our results about general decomposition along prioritization, it turns out that more complicated kinds of protection conditions arise. We thus find it useful to define the guarding of arbitrary pre-orders.

E.g, conjunctively decomposing when two minimized pre-orders are in parallel:

$$
\begin{aligned}
C(B; H1, H2; \emptyset; Z) \;\; \equiv \\
& B[Z] \\
& \wedge\; \neg\exists Z'.\, B[Z'] \wedge (Z' \prec_{H1} Z) \wedge (Z' \preceq_{H2} Z) \\
& \wedge\; \neg\exists Z'.\, B[Z'] \wedge (Z' \prec_{H2} Z) \wedge (Z' \preceq_{H1} Z)
\end{aligned}
$$

Intuitively, guards are a kind of half-way or one-directional fixture. Later, we will see how this is true in a precise sense.

Definition 5.6 (Guarding; Guarded Circumscription)

Let H and G be pre-orders defined over tuples similar to Z. Then we define a *guarded circumscription*, in which H is minimized while *guarding* G, to be:

$$C(B; min\ H; guard\, G; Z) \;\stackrel{\text{def}}{\equiv}\; B[Z] \;\wedge\; \neg\exists Z'.\, B[Z'] \wedge Z' \prec_H Z \;\wedge\; Z' \preceq_G Z$$

Note that the model theory for general circumscription generalizes straightforwardly to guarded circumscription.

Fixing Is a Special Case of Guarding:

Note that fixing is just a special case of guarding: fixing the pre-order $\preceq_F$ is equivalent to guarding the (equivalence) pre-order $\approx_F$.

Proposition 5.7 (Simple Case of Guarding: Parallel Predicates)

A simple case of guarding arises from conjunctively decomposing parallel predicate circumscription into single-predicate circumscription. Note that positivity of the base in the minimized predicates is *not* required here.

$$PPC(B; P^N; \emptyset; fix\ W; Z) \;\; \equiv$$

$$\bigwedge_{i \in N} PPC(B; Pi; \emptyset; fix\ W; guard\ P^{N-i}; Z)$$

Here, in each right-hand-side single-predicate circumscription, the guard pre-order is $(P^{N-i}; \emptyset)$, i.e., the conjoining of the rest of the predicates' pre-orders.

Proof : Special case of Theorem 5.14. □

Definition 5.8 (Guarded Policy)

We generalize the definition of the *policy* of a circumscription from Definition 2.8 to be: the specification of what pre-order is *guarded*, as well as of what pre-order is minimized, e.g., $(min\ H; guard\ G)$ above. We will speak of having *several guards*, perhaps including fixtures, in a circumscription. By this, we mean guarding a pre-order that is the conjunction, i.e., the parallel prioritization, of several guard (e.g., fixture) pre-orders.

Proposition 5.9 (Guarding Is Anti-Monotonic)

Adding a guard weakens any ordinary circumscription.
Adding an additional guard weakens any guarded circumscription.
Terminology: I.e., adding guard(s) is *anti-monotonic*.
Let $G1$, $G2$, and H be arbitrary pre-orders. Then

$$C(B;H;Z) \models C(B;H;guard\,G1;Z)$$

$$C(B;H;guard\,G1;Z) \models C(B;H;guard\,G1,G2;Z)$$

Proof : Very simple. □

Thus, adding a guard may prevent some NM conclusions from being sanctioned. A guard is a kind of limit or constraint on the non-monotonic effect of the minimization: it says that the minimization cannot be at the expense of increasing the (overall) guarded pre-order. Guarding a pre-order is not as strong a constraint as fixing it, however. A guard can be viewed as a half-way or one-sided fixture.

Proposition 5.10 (Guarding Versus Fixing)
Guarding a pre-order is less of a constraint than fixing that pre-order.

$$C(B;H;guard\,G;Z) \models C(B;H;fix\,G;Z)$$

Proof : Very simple. □

Ordinary Is Special Case of Guarded:
Note that ordinary circumscription is just a special case of guarded circumscription: in Definition 5.6, let G be $True$ (i.e., identically true), for example. In the remainder of this chapter, we will give some of our results for the case of guarded circumscription. All such results thus also apply to ordinary, non-guarded circumscription. In general, in decomposition, global guards, like global fixtures, are "inherited" by (i.e., are present in) each slice or phase policy.

Guarded Is a Special Case of Ordinary:
Our decomposition results below in this and the next section (e.g., Theorem 5.14) make use of guards. At first glance, this appears undesirable. What are these new beasts? Guarded circumscription, on its surface, is a more complex expressive class than ordinary circumscription, to which none of the substantial body of previous results apply directly. (By the way, most of the theorems discussed in this paper generalize to the case of guarded circumscriptions.) *Having* to resort to guarded circumscription feels perhaps artificial or clumsy. However, we show next that one can always reduce a guarded circumscription to an *ordinary* (i.e., non-guarded) circumscription in what is, mathematically, a fairly straightforward manner.

Actually, guarding is not a fundamental increase in the expressive complexity of general circumscription: every guarded policy is equivalent to an ordinary policy.

Observation 5.11 (Expressing Guards in the Minimized Pre-Order)

Every guarded policy is equivalent to an ordinary policy, in the sense of being equivalent for every base sentence; i.e., in the sense of forever equivalence (Definition 4.18) under base updates.

More precisely: Every guarded circumscription $C(B; min\ H; guard\ G; Z)$ is equivalent to any of a class of ordinary circumscriptions $C(B; min\ L; Z)$ that have the same base, where L obeys:

$$Z \prec_L Z' \ \equiv \ (Z \prec_H Z' \ \wedge \ Z \preceq_G Z')$$

E.g., one such L (the strongest in the class) is the reflexive closure of $\prec_L$:

$$Z \preceq_L Z' \ \stackrel{\text{def}}{\equiv} \ [(Z \prec_H Z' \ \wedge \ Z \preceq_G Z') \ \vee \ Z = Z']$$

Note that this forever equivalence to a *class* of (ordinary) policies, rather than to a unique policy, is the most we should hope for, since every *ordinary* policy H is forever equivalent to the class of ordinary policies $H2$ with the same strict version [2], i.e., such that:

$$Z \prec_{H2} Z' \ \equiv \ Z \prec_H Z'$$

Proof : By the definitions of ordinary and guarded general circumscription, the augmentation parts are the same. Thus the only question is whether there exists such L's that are well-defined as pre-orders. This is easily shown, e.g., for the reflexive closure, using basic properties of pre-orders plus predicate calculus manipulation. □

Issue: Preserving Expressive Class of the Policy:
However, we find the above method of expressively eliminating guards to be rather unnatural representationally for guards arising from decomposing prioritized predicate and default circumscriptions along prioritization, e.g., cf. Theorem 5.14 below. For the prioritized predicate and default classes of guarded policies (i.e., where both the guarded and minimized pre-orders are in that class), the above ordinary policy is not in the *same expressive class*. Thus more natural (to us) and satisfactory for our purposes is the following *method of guard reduction*, in which the ordinary policy *is* indeed in the same expressive class as the guarded.

By introducing new fixed predicates (GP below) and conservatively extending the base with axioms about them, one can always accomplish the effect of guarding an arbitrary pre-order. The intuition that a guard represents a kind of one-sided or half-way fixture turns out to be true in a precise sense.

Theorem 5.12 (Guard Reduction Method: General)
The guard(s) in any guarded circumscription can be replaced equivalently, in the sense of conservative extension, by: adding new predicates, fixing those predicates, and adding new base axiom(s) about them, as follows:

$$C(B; min\ H; guard\ G; Y) \ \equiv$$
$$\exists GP.\ C(B[Y] \wedge (Y \preceq_G GP); min\ H; fix\ GP; GP, Y)$$

Here the predicate tuple GP is similar to Y, and distinct from Y. H is applied to Y (not GP) in

[2] You may be wondering why bother with non-strict versions of pre-orders, if the circumscriptive theory does not care. One good reason is that the $\approx_H$ part of a pre-order $\preceq_H$ DOES indeed matter when aggregating pre-orders via prioritization.

the right-hand-side, ordinary circumscription. Note that the resulting ordinary policy is always in the same expressive class as the guarded policy.

Proof Overview: Non-trivial. Somewhat similar to the conservative extension theorem for abnormality theories (Theorem 3.8). Uses a generalization of positivity / negativity. See Appendix. □

Discussion of Guard Reduction Method:
The new fixed predicates GP are, in effect, place-holders. (GP is a mnemonic for "Guard Predicates".) The new base axiom(s) $Y \preceq_G GP$ conservatively extends B: no new conclusions just about the original predicates Y are entailed. Its effect is "transmit" the new fixture on GP so as to constrain the guard G one-sidedly (i.e., to prevent it from increasing) during the minimization of H. The second-order existential quantifier $\exists GP$ on the right-hand-side of the theorem just represents (i.e., achieves) the projection of the conservatively extended circumscriptive theory back onto the original predicates Y, i.e., the original base language. (Recall discussion of conservative extension in section 3.3.) Thus the guarded circumscription is equivalent to the GP-free part (see terminology after Definition 3.6) of the right-hand-side, ordinary circumscription.

5.4 Canonical Conjunctive Decomposition Along Prioritization

5.4.1 General Case

Equipped with the concept of guarding, we are ready to give our results about conjunctive decomposition on the basis of general, non-layered (external) prioritization.

Non-layered prioritization makes matters MUCH more complicated mathematically (just as it did in the very definition of prioritization).

We begin with some notation.

Definition 5.13 (Prioritization Partitions)
Let the index tuple N be the domain of a prioritization partial order R. For example, N indexes a tuple of starting pre-orders H, say, a tuple P of minimized predicates. Relative to that prioritization partial order R, we define, for each index $i \in N$, the partition of the remainder of N into the indices that are strictly higher priority than (Dominate) i, those that are strictly lower priority than (Dominated By) i, and those that are neither higher nor lower priority than (Incomparable to) i.

$$R_D(i) \stackrel{\text{def}}{=} \{j | R(j,i)\}$$
$$R_{DB}(i) \stackrel{\text{def}}{=} \{j | R(i,j)\}$$
$$R_I(i) \stackrel{\text{def}}{=} \{j | \neg R(i,j) \wedge \neg R(j,i) \wedge j \neq i\}$$

Theorem 5.14 (Conjunctive, Pre-Orders, with Guarding)

Any prioritized circumscription, or, more generally, any guarded prioritized circumscription, can be conjunctively decomposed into *guarded* circumscriptions in which a single of its starting pre-orders is minimized, as follows:

$$C(B; min\ (H^N; R); fix\ F; guard\ G; Z) \quad \equiv$$

$$\bigwedge_{i \in N} C(B; min\ Hi; fix\ H^{R_D(i)}, F; guard\ (H^{R_I(i)}; R^{R_I(i)}), G; Z)$$

Proof Overview: Complicated. Uses several lemmas about prioritized pre-orders. See Appendix. $\square$

Theorem 5.15 (Conjunctive, Pre-Orders, Reduced Guards)
Any prioritized circumscription can be conjunctively decomposed along its prioritization into *ordinary* circumscriptions in each of which a single of its starting pre-orders is minimized, as follows:

$$C(B; min\ (H^N; R); Y) \quad \equiv$$

$$\bigwedge_{i \in N} \exists GPTi.\ C(B \wedge (Y \preceq_{(H^{R_I(I)}; R^{R_I(i)})} GPTi)\ ; min\ Hi; \\ fix\ GPTi, H^{R_D(i)}; GPTi, Y)$$

which is also equivalent to:

$$\exists GPT.\ \bigwedge_{i \in N} C(B \wedge (Y \preceq_{(H^{R_I(I)}; R^{R_I(i)})} GPTi)\ ; min\ Hi; \\ fix\ GPTi, H^{R_D(i)}; GPT, Y)$$

Here, for each $i \in N$, $GPTi$ is a distinct tuple of new predicates that is similar to Y. (Each $GPTi$ is distinct both from Y and from every other $GPTj$ for $j \neq i$.) GPT stands for the tuple of the $GPTi$'s. As in Theorem 5.12, Hi on the right-hand-side is applied to Y, not to $GPTi$.

Proof : Immediate from Theorem 5.14 plus Theorem 5.12. $\square$

Discussion:
In Theorems 5.14 and 5.15, the introduced fixtures and guards in each slice can be viewed as extra "interaction terms" that protect against over-stepping in situations of conflicting interactions with the external context. The interaction terms depend on relative prioritization. (By "terms" here, we mean in the sense of polynomials, e.g., in statistical or engineering models, not in the sense usual in logic of a functional expression.)

We postpone further intuitive interpretation until the next subsection, where we discuss the predicate case.

Definition 5.16 (Canonical Conjunctive Decomposition)
We say that the conjunctive decomposition cf. Theorems 5.14 and 5.15 is *canonical*. It can always be applied. And, as we will discuss later in sections 5.8 and 5.12, it is useful for defining various notions such as: 1) canonical derivability from a subset of defaults; and 2) the set of previous higher-priority conclusions, relative to a default update.

Definition 5.17 (Canonical Protection Tactic)
By the *canonical protection tactic*, we mean the method of introducing fixtures and guards in the decomposition along prioritization cf. Theorem 5.14. We also apply the term to describe the *reduced form* of the protection in Theorem 5.15. We also speak of the constituent axiom sets being canonical, both in the guarded form and in the ordinary (reduced) form.

5.4.2 Predicates, One by One

Next, we give the predicate case of canonical conjunctive decomposition cf. Theorems 5.14 and 5.15.

Corollary 5.18 (Conjunctive, General Priority: Predicates)
Any prioritized predicate circumscription can be conjunctively decomposed along its prioritization into *ordinary* single-predicate circumscriptions, and also into *guarded* single-predicate circumscriptions, using the canonical protection tactic, as follows:
I) Guarded Form: The predicate special case of Theorem 5.14 implies that:

$$PPC(B; P; R; fix\ W; Y) \equiv$$
$$\bigwedge_{i \in N} PPC(B; P_i; \emptyset; fix\ P^{R_D(i)}, W; guard\ (P^{R_I(i)}; R^{R_I(i)}); Y)$$

II) Guard Reduction: We next apply, to each guarded single-predicate circumscription above, the prioritized-predicate special case of guard reduction cf. Theorem 5.12, which we give in general form:

$$PPC(B; P; R; fix\ W; guard\ (Q; R2); Y) \equiv$$
$$\exists GP.\ PPC(B[Y] \wedge (Q \preceq_{R2} GP); fix\ GP, W; GP, Y)$$

where the predicate tuples Q, P, and W are distinct. Note that GP is similar to Q, rather than to all of Y.
III) Ordinary Form (After Guard Reduction): Combining **I)** and **II)** yields:

$$PPC(B; P^N; R; fix\ W; Y) \equiv$$
$$\bigwedge_{i \in N} \exists GPTi.\ PPC(B \wedge (P^{R_I(i)} \preceq_{R^{R_I(i)}} GPTi)\ ; Pi; \emptyset;$$
$$fix\ GPTi, P^{R_D(i)}, W; GPTi, Y)$$

which is also equivalent to:

$$\exists GPT.\ \bigwedge_{i \in N} PPC(B \wedge (P^{R_I(i)} \preceq_{R^{R_I(i)}} GPTi)\ ; Pi; \emptyset;$$
$$fix\ GPTi, P^{R_D(i)}, W; GPT, Y)$$

Here, for each $i \in N$, $GPTi$ is a distinct tuple of new predicates that is similar to $P^{R_I(i)}$. (Each $GPTi$ is distinct both from Y and from every other $GPTj$ for $j \neq i$.) GPT stands for the tuple of the $GPTi$'s. (Equivalently, the fixing of $GPTi$ in the second equivalence can be replaced by the fixing of GPT.)

Proof : Immediate as special case of Theorems 5.14 and 5.15. □

Intuition: Protection and Turf:
Corollary 5.18, and, more generally, Theorems 5.14 and 5.15, can be viewed intuitively in terms of protecting "turf" with a pecking order corresponding to priority. In the slice where a predicate Pi is minimized, it must "stay away" from the higher-priority predicates $P^{R_D(i)}$: they are fixed, and thus (when immunity of the fixed holds — see Theorem 3.39) no new conclusions will be made about them. However, the lower-priority predicates $P^{R_{DB}(i)}$ are permitted to vary freely: their turf

is "up for grabs". So far, all this is as in Lifschitz' previous result for the layered predicate case (Theorem 5.1).

Incomparable Are Guarded:
A major difference from Theorem 5.1 is the consideration of incomparability with respect to prioritization. Incomparability arises from *partial*-ness of the prioritization partial order. The incomparable predicates $P^{R_I(i)}$ are *guarded*: their turf is available, but only for "co-operation": no increase, going against their minimization, is permitted. Moreover, there is a subtlety: this co-operation condition must take into account the *internal prioritization amongst the incomparable* predicates: $R^{R_I(i)}$.

Intuitive Interpretation of Reduced Guards:
As in the general case of guard reduction (Theorem 5.12), the new fixed predicates $GPTi$ in each slice are, in effect, place-holders. ($GPTi$ is a mnemonic for "Guard Predicates Tuple, slice i".) The new base axiom $GPTi \preceq_{R^{R_I(i)}} P^{R_I(i)}$ conservatively extends B: no new conclusions just about the original predicates Y are entailed. Its effect is to "transmit" the new fixture on $GPTi$ so as to constrain the incomparable predicates $P^{R_I(i)}$ one-sidedly, i.e., to prevent them from increasing, modulo their internal prioritization $R^{R_I(i)}$, during the minimization of Pi. The second-order existential quantifier $\exists GPTi$ on the right-hand-side of the theorem just represents (i.e., achieves) the projection of the conservatively extended circumscriptive theory back onto the original predicates Y, i.e., the original base language. (Recall discussion of conservative extension in section 3.3.) Thus the guarded slice is equivalent to the $GPTi$-free part (see terminology after Definition 3.6) of the ordinary slice with the new fixtures and the new base axioms about them.

Protection Tactic Guards and Fixtures As Declaration Axioms:
One can view the introduced guards and fixtures as extra quasi-axioms, i.e., as *"declarations"* summarizing the relevant external context that bears on the defaults being maximized in the slice. This is analogous to the role of external declarations in programming languages with side-effects, as we discuss more in section 5.6.

Interestingly, as far as each slice is concerned, only some of the other globally-minimized predicates are relevant: the **lower-priority are irrelevant** (irrelevant in the sense discussed in Observation 4.8) for *any* prioritization partial order, not just for layered prioritization partial orders, as in Theorem 5.1.

Example 5.19 (Meetings)
Consider the Meetings example from sections 2.6 and 2.8. Let the base sentence B consist of $B1$ from section 2.8 conjoined with additional axioms E that assert some specific for-sure information about the various predicates mentioned there, i.e., that particular tuples are, and others are not, fires, vacations, Tuesdays, weekdays, sicknesses, holidays, workings, etc.; these additional axioms E (e.g., $sick(Ed, Today)$ in section 2.8) produce some conflicts among the defaults.

We represent the non-monotonic theory via the prioritized predicate circumscription $PPC(B; ab; R; Z)$, where ab, R, and Z are as in section 2.8.

Then, for any base B (e.g., for any E and associated conflicts), this circumscription is equivalent to its canonical conjunctive decomposition along its global prioritization, cf. form **III)** in Corollary 5.18. This decomposition has six slices, one per minimized predicate (i.e., one per default). Next, we define that decomposition.

Let the new predicate symbol $gpij$ stand for $GPTij$ in Corollary 5.18. Let the base sentence for the i^{th} slice be:

$SBi \stackrel{\text{def}}{\equiv} B \wedge (ab^{R_I(i)} \preceq_{R^{R_I(i)}} gpi)$

where gpi is the tuple of the $gpij$'s.

For the R in this example:

$R_I(1) = \emptyset \qquad R_D(1) = \emptyset$

$R_I(2) = \{4,5,6\} \qquad R_D(2) = \{1\}$

$R_I(3) = \{4,5,6\} \qquad R_D(3) = \{1,2\}$

$R_I(4) = \{2,3,5\} \qquad R_D(4) = \{1\}$

$R_I(5) = \{2,3,4\} \qquad R_D(5) = \{1\}$

$R_I(6) = \{2,3\} \qquad R_D(6) = \{1,4,5\}$

Thus, according to Corollary 5.18:

$PPC(B; ab; R; Z)$

is equivalent to:

$$\bigwedge_i \exists gpi.\ PPC(SBi; abi; \emptyset; fix\ gpi, ab^{R_D(i)}; Z, gpi)$$

which, elaborating, is defined as:

$\exists gp1.\ PPC(SB1; ab1; \emptyset; Z)\ \wedge$

$\exists gp2.\ PPC(SB2; ab2; \emptyset; fix\ gp2, ab1; Z, gp2)\ \wedge$

$\exists gp3.\ PPC(SB3; ab3; \emptyset; fix\ gp3, ab1, ab2; Z, gp3)\ \wedge$

$\exists gp4.\ PPC(SB4; ab4; \emptyset; fix\ gp4, ab1; Z, gp4)\ \wedge$

$\exists gp5.\ PPC(SB5; ab5; \emptyset; fix\ gp5, ab1; Z, gp5)\ \wedge$

$\exists gp6.\ PPC(SB6; ab6; \emptyset; fix\ gp6, ab1, ab4, ab5; Z, gp6)$

where:

$SB1 \stackrel{\text{def}}{\equiv} B$

$SB2 \stackrel{\text{def}}{\equiv} B \wedge ab^{456} \preceq_{R^{456}} gp2$

$\quad \stackrel{\text{def}}{\equiv} B \wedge (ab4 \leq gp21) \wedge (ab5 \leq gp22)$

$\qquad \wedge [(ab4{=}gp21) \wedge (ab5{=}gp22) \supset (ab6 \leq gp23)]$

(Here in superscripts, we let 456 stand for the tuple $\langle 4,5,6 \rangle$, as shorthand. Note that $gp2$ is similar to ab^{456}. Continuing, in like fashion:)

$SB3 \stackrel{\text{def}}{\equiv} B \wedge ab^{456} \preceq_{R^{456}} gp3$

$\quad \stackrel{\text{def}}{\equiv} B \wedge (ab4 \leq gp31) \wedge (ab5 \leq gp32)$

$\qquad \wedge [(ab4{=}gp31) \wedge (ab5{=}gp32) \supset (ab6 \leq gp33)]$

$SB4 \stackrel{\text{def}}{\equiv} B \wedge ab^{235} \preceq_{R^{235}} gp4$

$\quad \stackrel{\text{def}}{\equiv} B \wedge (ab2 \leq gp41)$

$$
\begin{aligned}
& \qquad\qquad\quad \wedge\, [(ab2{=}gp41) \supset (ab3 \leq gp42)] \\
& \qquad\qquad\quad \wedge\, (ab5 \leq gp43) \\
SB5 \ &\overset{\text{def}}{\equiv}\ B \ \wedge\ ab^{234} \preceq_{R^{234}} gp5 \\
&\overset{\text{def}}{\equiv}\ B \ \wedge\ (ab2 \leq gp51) \\
& \qquad\qquad\quad \wedge\, [(ab2{=}gp51) \supset (ab3 \leq gp52)] \\
& \qquad\qquad\quad \wedge\, (ab4 \leq gp53) \\
SB6 \ &\overset{\text{def}}{\equiv}\ B \ \wedge\ ab^{23} \preceq_{R^{23}} gp6 \\
&\overset{\text{def}}{\equiv}\ B \ \wedge\ (ab2 \leq gp61) \\
& \qquad\qquad\quad \wedge\, [(ab2{=}gp61) \supset (ab3 \leq gp62)]
\end{aligned}
$$

Note that, in effect, some of the expressive complexity of the non-layered prioritization has been "moved" into the slices' bases. E.g., even if B is universal (Definition 3.27), $SB2$ is not. Later (e.g., in section 5.11), we will return to this question of the expressive complexity of the canonical conjunctive decomposition.

Corollary 5.20 (Conjunctive, Parallel Predicates Without Positivity)
In the parallel special case of Corollary 5.18, the reduced guards have especially simple form. Any parallel predicate circumscription can be conjunctively decomposed into *ordinary* single-predicate circumscriptions, using the canonical protection tactic, as follows:

$$PPC(B; P^N; \emptyset; fix\ W; Y) \ \equiv$$

$$\bigwedge_{i \in N} \exists GPTi.\ PPC(B \wedge (P^{N-i} \leq GPTi)\ ; Pi; \emptyset; fix\ GPTi, W; GPTi, Y)$$

or into *guarded* single-predicate circumscriptions cf. Proposition 5.7.

Proof : Immediate as special case of Corollary 5.18. □

Note that for this parallel predicate case, one can achieve a similar result by combining conservative extension for adaptive abnormality (Theorem 3.8) with Theorem 5.3, i.e., from Lifschitz' previous results. Our mathematical contribution for conjunctive decomposition along prioritization is primarily for the non-empty, and, especially, non-layered case of (global) prioritization.

5.4.3 Defaults, One by One

Next, we give the default case of canonical conjunctive decomposition cf. Theorems 5.14 and 5.15.

Corollary 5.21 (Conjunctive, General Priority, Defaults)
Any prioritized default circumscription can be conjunctively decomposed along its prioritization into *ordinary* single-default circumscriptions, or into *guarded* single-default circumscriptions, using the canonical protection tactic, as follows:

I) Guarded Form: The default special case of Theorem 5.14 implies that:

$$PDC(B; D^N; R; fix\ F; Y) \ \equiv$$

$$\bigwedge_{i \in N} PDC(B; Di; \emptyset; fix\ D^{R_D(i)}, F; guard\ (D^{R_I(i)}; R^{R_I(i)}); Y)$$

Notation: maximized guards: In $PDC(\ldots)$ notation, guards are maximized, rather than minimized as in $C(\ldots)$ and $PPC(\ldots)$ notation. I.e., the guard pre-orders' directionality is treated similarly to the main minimized / maximized pre-order's, in each notation.

II) Guard Reduction: We next apply, to each guarded single-default circumscription above, the prioritized-default special case of guard reduction cf. Theorem 5.12, which we give in general form.

Let $(G; R2)$ be a prioritized-default pre-order guard. Then

$$PDC(B; D; R; \mathit{fix}\ F; \mathit{guard}\ (G; R2); Y) \quad \equiv$$
$$\exists GP.\ PDC(B[Y] \wedge (G[GP] \preceq_{R2} G[Y])\ ; D; R; \mathit{fix}\ GP, F; GP, Y)$$

where GP is a tuple of new predicates that is similar to Y.

Note that, actually, one need only introduce new predicates that are mentioned in G; thus GP may be similar to a subset, rather than all, of Y: the subset mentioned in $G[Y]$.

III) Ordinary Form (After Guard Reduction): Combining **I)** and **II)** yields:

$$PDC(B; D^N; R; \mathit{fix}\ F; Y) \quad \equiv$$

$$\bigwedge_{i \in N} \exists GPTi.\ PDC(B \wedge (D^{R_I(i)}[GPTi] \preceq_{R^{R_I(i)}} D^{R_I(i)}[Y])\ ; Di; \emptyset;$$
$$\mathit{fix}\ GPTi, D^{R_D(i)}, F; GPTi, Y)$$

which is also equivalent to:

$$\exists GPT.\ \bigwedge_{i \in N} PDC(B \wedge (D^{R_I(i)}[GPTi] \preceq_{R^{R_I(i)}} D^{R_I(i)}[Y])\ ; Di; \emptyset;$$
$$\mathit{fix}\ GPTi, D^{R_D(i)}, F; GPT, Y)$$

Here, $GPTi$ and GPT are as in the predicate case (Corollary 5.18) except that $GPTi$ is similar to $D^{R_I(i)}[Y]$ instead. Note that in the right-hand-side circumscriptions, Di, $D^{R_D(i)}$, and F are applied to Y, not to $GPTi$, as in the guard reduction Theorem 5.12.

Each PDC in this ordinary-form conjunctive decomposition corresponds to a CLD axiom set.

IV) In CLD: Any global CLD axiom set can be conjunctively decomposed, in the sense of Definition 4.2, along its prioritization in the above fashion **III)**. As above, let N index the defaults. Let n be the number of defaults, i.e., the size of N. Then there are n constituent axiom sets $\mathcal{SA}_i$, one for each $i \in N$. Each $\mathcal{SA}_i$ contains:

- exactly one default axiom: $:>\ Di$
- none of the other global default axioms
- no prioritization axioms
- the global base axioms
- the global fixture axioms
- fixture axioms asserting the fixing of each new guard predicate in the tuple $GPTi$
- base axioms asserting the (reduced guard) relationship:
  $D^{R_I(i)}[GPTi] \preceq_{R^{R_I(i)}} D^{R_I(i)}[Y]$

In addition, in the first form of the decomposition equivalence in **III)** above, conjunctive combination is performed using only the $GPTi$-free part of each slice; in the second form, only the GPT-free part of the conjunctive combination is used.

Proof : Immediate as special case of Theorems 5.14 and 5.15. □

Intuition: Protection, Turf, Guards:
Corollary 5.21, as in the predicate case (see discussion after Corollary 5.18), can be viewed intuitively in terms of protecting "turf" with a pecking order corresponding to priority. The differences from the predicate case are that: default formulas play the role of predicates; and they are being maximized rather than minimized. In the slice where a default Di is *maximized*, it must "stay away" from the higher-priority defaults' *default formulas* $D^{R_D(i)}$: they are fixed; however, the lower-priority defaults' default formulas $D^{R_{DB}(i)}$ are "up for grabs". That is, the **lower-priority defaults are irrelevant** in the sense discussed in Observation 4.8. The incomparable defaults' default formulas $D^{R_I(i)}$ are *guarded*: their turf is available, but only for "co-operation": no decrease, going against their maximization, is permitted. More precisely, this co-operation condition takes into account the *internal prioritization amongst the incomparable* defaults: $R^{R_I(i)}$. In the guard reduction, the fixing of the newly introduced guard predicates constrains the incomparable defaults' default formulas one-sidedly, i.e., to prevent them from *decreasing*.

Example 5.22 (Meetings)
Example 5.19 is easily adapted to a prioritized default circumscription: replace all the abnormality predicates by the corresponding default formulas, in the manner of Theorem 3.8. I.e., replace $ab1$ by $fire(d,t) \supset leave(p,d,t)$, etc.. Then the canonical conjunctive decomposition of this default-form example looks similar to the predicate-form: there are six slices, one per default.

Example 5.23 (Inheritance of skin properties)
As a simpler example, we elaborate upon the inheritance example illustrated in Figure 2.4 in subsection 2.7.3.

Let Y stand for the tuple of predicates
$\langle dolphin, hairy\_skin, mammal, albino\_elephant, white\_skin, elephant, gray\_skin\rangle$

Let the default formulas $D[Y]$ be:

$$D1[Y,x] \stackrel{\text{def}}{\equiv} dolphin(x) \supset \neg hairy\_skin(x)$$
$$D2[Y,x] \stackrel{\text{def}}{\equiv} mammal(x) \supset hairy\_skin(x)$$
$$D3[Y,x] \stackrel{\text{def}}{\equiv} albino\_elephant(x) \supset white\_skin(x)$$
$$D4[Y,x] \stackrel{\text{def}}{\equiv} elephant(x) \supset gray\_skin(x)$$

The prioritization, based on specificity dominance, is columnar (Definition 2.54). Let the precedence partial order R be the relation containing exactly the pairs $(1,2)$ and $(3,4)$.

Let the global base formula $B[Y]$ be the conjunction of:

$$\forall x.\ dolphin(x) \supset mammal(x)$$
$$\forall x.\ albino\_elephant(x) \supset elephant(x)$$

$\forall x.\ \neg(white\_skin(x) \wedge gray\_skin(x))$

together with some additional axioms about which particular objects are, and which are not: dolphins, mammals, elephants, albino elephants, hairy-skinned, white-skinned, gray-skinned; these additional axioms produce some conflicts among the defaults. E.g.:

$dolphin(Flipper)$

$albino\_elephant(Clyde)$

$elephant(Dumbo)$

$mammal(Ping)$

$mammal(Pong)$

$\neg hairy(Ping) \vee \neg hairy(Pong)$

$\forall x.\ elephant(x) \supset mammal(x)$

We represent the non-monotonic theory via the prioritized default circumscription $PDC(B;D;R;Y)$.

Then this circumscription is equivalent to its canonical conjunctive decomposition along its global prioritization, cf. form **III)** in Corollary 5.21. This decomposition has four slices, one per default. Next, we define that decomposition.

Let the new predicate symbol $gpij$ stand for $GPTij$ in Corollary 5.21. Let the base sentence for the i^{th} slice be:

$$SBi \ \stackrel{\text{def}}{\equiv}\ B \ \wedge\ (D^{R_I(i)}[gpi] \preceq_{{}_R R_I(i)} (D^{R_I(i)}[Y])$$

where gpi is the tuple of the $gpij$'s.

For the R in this example:

$R_I(1) = \{3,4\}$ $\qquad R_D(1) = \emptyset$

$R_I(2) = \{3,4\}$ $\qquad R_D(2) = \{1\}$

$R_I(3) = \{1,2\}$ $\qquad R_D(3) = \emptyset$

$R_I(4) = \{1,2\}$ $\qquad R_D(4) = \{3\}$

Thus, according to Corollary 5.21:

$PDC(B;D;R;Y)$

is equivalent to:

$$\bigwedge_i \exists gpi.\ PDC(SBi;Di;\emptyset;fix\ gpi, D^{R_D(i)};Y,gpi)$$

which, elaborating, is defined as:

$$\begin{aligned}
&\exists gp1.\ PDC(SB1;D1;\emptyset;fix\ gp1;Y,gp1)\ \wedge\\
&\exists gp2.\ PDC(SB2;D2;\emptyset;fix\ gp2,D1;Y,gp2)\ \wedge\\
&\exists gp3.\ PDC(SB3;D3;\emptyset;fix\ gp3;Y,gp3)\ \wedge\\
&\exists gp4.\ PDC(SB4;D4;\emptyset;fix\ gp4,D3;Y,gp4)\ \wedge
\end{aligned}$$

where each gpi is similar to Y, and:

$$\begin{aligned}
SB1 \ &\stackrel{\text{def}}{\equiv}\ B \ \wedge\ D^{34}[gp1] \preceq_{R^{34}} D^{34}[Y]\\
&\stackrel{\text{def}}{\equiv}\ B \ \wedge\ (D3[gp1] \leq D3[Y])\\
&\qquad\qquad \wedge [(D3[gp1]{=}D3[Y]) \supset (D4[gp1] \leq D4[Y])]
\end{aligned}$$

(Here in superscripts, we let 34 stand for the tuple $\langle 3,4\rangle$, as shorthand. Continuing, in like fashion:)

$$
\begin{aligned}
SB2 &\stackrel{\text{def}}{\equiv} B \ \wedge\ D^{34}[gp2] \preceq_{R^{34}} D^{34}[Y] \\
&\stackrel{\text{def}}{\equiv} B \ \wedge\ (D3[gp2] \leq D3[Y]) \\
&\qquad\qquad \wedge [(D3[gp2]{=}D3[Y]) \supset (D4[gp2] \leq D4[Y])] \\
SB3 &\stackrel{\text{def}}{\equiv} B \ \wedge\ D^{12}[gp3] \preceq_{R^{12}} D^{12}[Y] \\
&\stackrel{\text{def}}{\equiv} B \ \wedge\ (D1[gp3] \leq D1[Y]) \\
&\qquad\qquad \wedge [(D1[gp3]{=}D1[Y]) \supset (D2[gp3] \leq D2[Y])] \\
SB4 &\stackrel{\text{def}}{\equiv} B \ \wedge\ D^{12}[gp4] \preceq_{R^{12}} D^{12}[Y] \\
&\stackrel{\text{def}}{\equiv} B \ \wedge\ (D1[gp4] \leq D1[Y]) \\
&\qquad\qquad \wedge [(D1[gp4]{=}D1[Y]) \supset (D2[gp4] \leq D2[Y])]
\end{aligned}
$$

Elaborating yet more explicitly, to give the flavor of what the above actually says in terms of the primitive predicate symbols:

$$
\begin{aligned}
SB3 \stackrel{\text{def}}{\equiv} B \wedge \\
&[\forall x.\ (gp31(x) \supset \neg gp32(x)) \supset (dolphin(x) \supset \neg hairy\_skin(x))] \ \wedge \\
&\{[\forall x.\ (gp31(x) \supset \neg gp32(x)) \equiv (dolphin(x) \supset \neg hairy\_skin(x))] \\
&\qquad \supset\ [\forall x.\ (gp33(x) \supset gp34(x)) \supset (mammal(x) \supset hairy\_skin(x))]\}
\end{aligned}
$$

Corollary 5.24 (Conjunctive, Parallel Defaults)
Any parallel default circumscription can be conjunctively decomposed into *ordinary* or *guarded* single-default circumscriptions, as follows:

$$PDC(B; D^N; \emptyset; fix\ F; Y)$$

is equivalent to the guarded form:

$$\bigwedge_{i \in N} PDC(B; Di; \emptyset; fix\ F; guard\ D^{N-i}; Y)$$

(Here, as in the parallel predicate case, we notationally omit the empty prioritization of the guard pre-order.)
This is, in turn, equivalent to the ordinary-form:

$$\bigwedge_{i \in N} \exists GPTi.\ PDC(B \wedge (D^{N-i}[GPTi] \leq D^{N-i}[Y])\ ; Di; \emptyset; fix\ GPTi, F; Y)$$

Proof : Immediate as special case of Corollary 5.21. □

5.4.4 Hierarchy and Modules

In this subsection, we show that one can canonically conjunctively decompose (cf. Theorems 5.14 and 5.15) circumscriptive *theories* recursively or *hierarchically* via decomposition of the *prioritization* partial order: breaking an original global circumscription into several slices, then breaking each of those slices into finer slices, and so on. *One can pick the grain size and hierarchical scope of the decomposition.*

Guide to Reader: Recall the introduction of the idea of modules in Definition 2.62 and subsection 2.9.3, before continuing with the rest of this subsection.

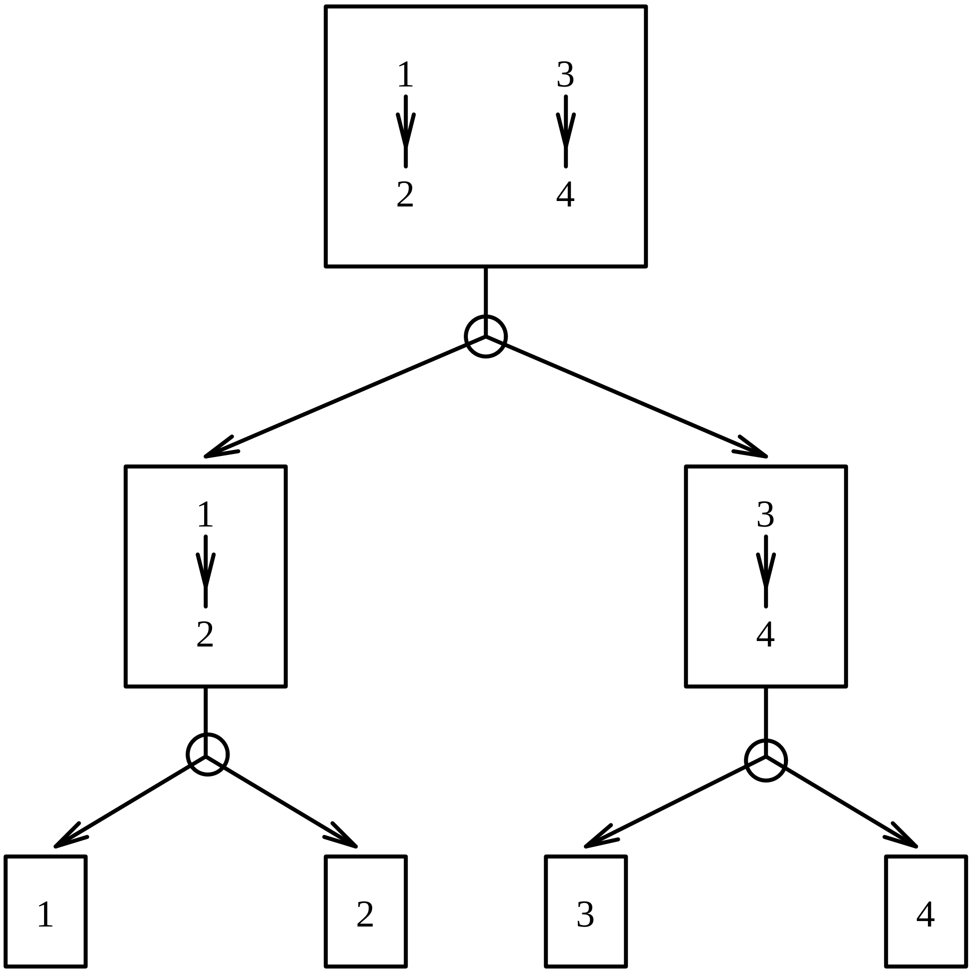


Figure 5.1: Decomposition by Modules: Skin-Properties Inheritance example.

In this hierarchical decomposition, each of the decomposition stages corresponds to decomposing a module into its sub-modules. In each intermediate-grain-size slice, the defaults and their prioritization correspond to an intermediate-grain-size module. Each intermediate-grain-size module contains fewer defaults than the global, but typically contains more than one default.

Example 5.25 (Decomposition by Modules: Skin-Properties Inheritance)
In the skin-properties inheritance example (Figure 2.4 and Example 5.23), in which the global prioritization is columnar (Definition 2.54), there are two columns in parallel: one with two defaults about skin hairiness; and the other with two defaults about skin color. Each column corresponds to a column module. Thus in Example 5.23, one can canonically conjunctively decompose the global into two slices, one per column module. In each column slice, the only defaults are those contained in that column. Each column slice can then in turn be canonically conjunctively decomposed into one slice per default, i.e., one slice per primitive module. Figure 5.1 illustrates.

Example 5.26 (Decomposition by Modules: Meetings)

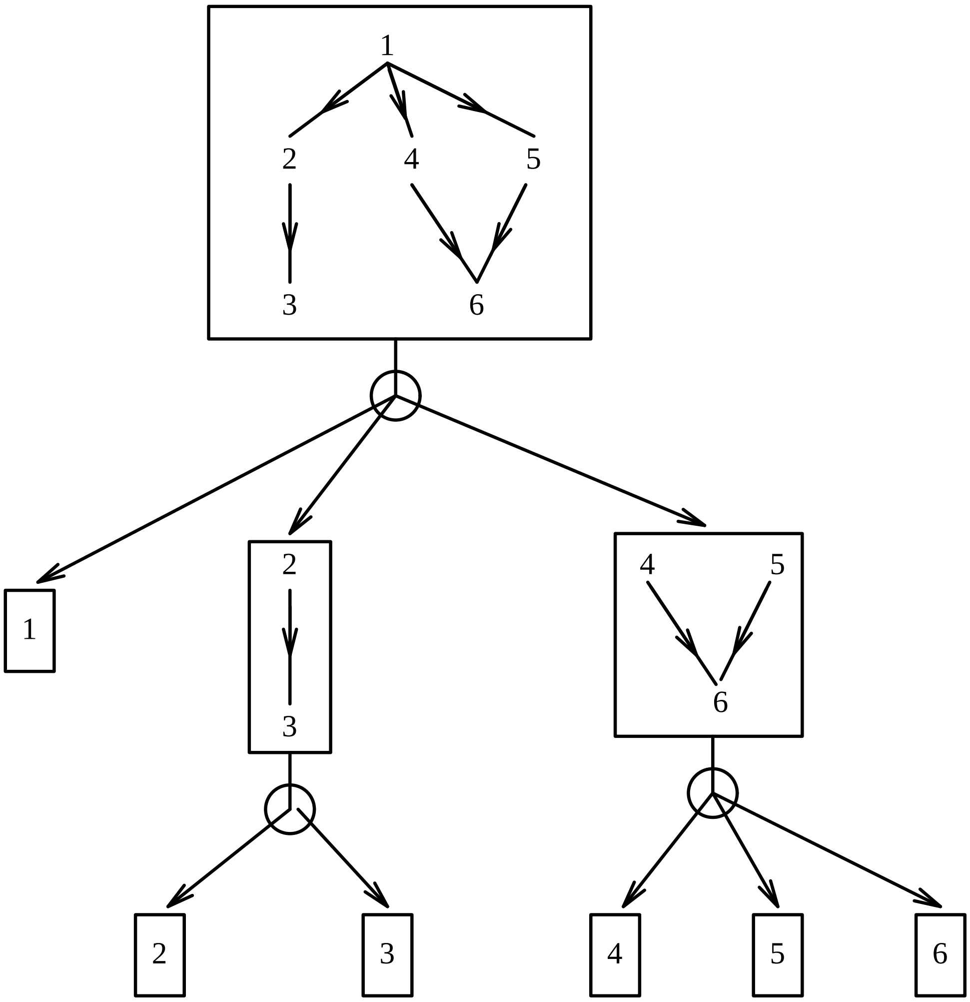


Figure 5.2: Decomposition by Modules: Meetings example.

A bit more complexly, the Meetings example (Figure 2.1, sections 2.6 and 2.8, Example 5.22) can be canonically conjunctively decomposed into three slices, one per module: i.e., one for "emergencies", one for "meetings", and one for "attendance", in terms of Figure 2.1. Each of these three module slices can then in turn be canonically conjunctively decomposed into one slice per default. Figure 5.2 illustrates.

Next, we give all this in more precise detail.

Definition 5.27 (Module Circumscription)

We define a *module* circumscription $MC(B; H; fix\ F; Z)$ as the circumscriptive *maximization* of a module pre-order H. Recall that every module pre-order/circumscription $H \stackrel{\text{def}}{=} (SH; R)$ is a prioritized default pre-order/circumscription.

Notation: Maximizing Circumscription: More generally, we write MC to indicate that a circumscription is maximizing rather than minimizing. As with $PDC(\ldots)$ notation, in $MC(\ldots)$

notation, guards are maximized as well.

To show how to perform hierarchical decomposition, we need only show how to perform one recursive step of decomposition. Our next result gives that recursive step.

Corollary 5.28 (Conjunctive, Modules into Sub-Modules)
Any module circumscription can be conjunctively decomposed into *ordinary* single-sub-module circumscriptions (where the module is equivalent to the prioritization of those sub-modules), and also into *guarded* single-sub-module circumscriptions, using the canonical protection tactic, as follows.

I) Guarded Form: In Corollary 2.63, let NE stand for the index tuple of RE, i.e., of the (sub-)modules tuple MOD.

Then

$MC(B; H; fix\ F; Z) \quad \equiv$

$$\bigwedge_{j \in NE} MC(B; MODj; fix\ MOD^{RE_D(j)}, F; guard\,(MOD^{RE_I(j)}; RE^{RE_I(j)}); Z)$$

II) Ordinary CLD Form: The above guarded form (circumscription) can be reduced to ordinary form via the method of Theorem 5.12.

This ordinary form can be expressed in CLD, since each (sub-)module pre-order, whether maximized, fixed, or guarded, is a prioritized default pre-order. There is a subtlety, however, in expressing the fixed modules, however; see point (**) below.

In terms of CLD: a global (module) CLD axiom set (circumscription) can be canonically decomposed into a conjunction of (sub-)module slices (circumscriptions), where the slice axiom set for module j includes:

- exactly the defaults, and the prioritization among them, from module j
- none of the remainder of the global set of defaults;
  i.e., no defaults from the other modules
- all the global base axioms
- all the global fixture axioms
- fixture axioms that fix the default formulas of all of the defaults in the higher-priority (than j, in the inter-module prioritization RE) modules;
  (**) note that this is equivalent to fixing the higher-priority modules' pre-orders ($MOD^{RE_D(j)}$)
  (see the lemma mentioned in the proof overview below)
- base and fixture axioms that accomplish, via the guard reduction method in Corollary 5.21, the guarding of the default formulas of all of the defaults in the incomparable-priority (than j, in the inter-module prioritization RE) modules;
  note that the internal prioritization involved in guarding these default formulas is the (parallel composition of) the internal prioritization of those incomparable-priority modules

- no protection with respect to the lower-priority (than j, in the inter-module prioritization RE) modules

Proof : Immediate as special case of Theorems 5.14 and 5.15, except for the equivalence (**) above. Lemma B.10 (see Appendix) says that fixing any prioritized pre-order ($G^{NG}; RG$) is equivalent to fixing all of the starting pre-orders G^{NG} (in parallel). In particular, it implies that fixing a module pre-order is equivalent to fixing all of the defaults in it. □

Discussion:
The above corollary can be applied recursively / hierarchically, exploiting the composition of prioritization, i.e., of modules, as a basis for selecting the grain size of the slices. I.e., the decomposition of the global prioritization partial order can be exploited as a basis for conjunctive decomposition of the circumscription. In each step of canonical decomposition, each sub-module slice has fewer defaults and simpler prioritization than the super-module. However, it involves more stuff to protect: it has more external context of higher- and incomparable- priority defaults that need to be fixed or guarded. A perspective is that all of the defaults in a given module share a common set of protection conditions, since they share a common external context (i.e., the rest of the modules).

Note that our earlier results on canonical conjunctive decomposition one-by-one are special cases of the above corollary. The one-by-one case is when the global module is viewed as the prioritization of primitive sub-modules. Note that this finest-grain slicing involves more stuff to protect than coarser-grain slicings.

Corollary 5.28 generalizes straightforwardly to decomposing a guarded module circumscription, e.g., arising from an earlier decomposition step. The initial guard is then just "carried along": it appears in each of the slices, just as G does in in Theorem 5.14. In short, Corollary 5.28 generalizes straightforwardly to repeatedly decomposing with guarded forms.

5.4.5 Finest Grain: Instances

Next, we show that hierarchical canonical decomposition can, often, produce micro-grain, as well as macro-grain, canonical decomposition.

Theorem 5.29 (Instance-Grain Decomposition)
Note that if a PDC is quasi-propositional (Definition 3.22), e.g. if the base includes domain closure, then the set of default instances for each default axiom is finite and countable. (See Definition 2.46 for the definition of a default instance.) In this case, one may canonically decompose one-by-one to finest instance-grain so that there is just one *closed* default *instance* in each slice, even when some or all of the global default axioms are open. This is because each open default axiom can be viewed as a module formed by prioritizing in parallel the tuple of all that default's (closed) default instances. (See Theorem 2.47 and Lemma 3.23) Thus each slice with an open default axiom $Di[Y,x]$ (e.g., $Quaker(x) \supset Pacifist(x)$) can be canonically sliced further into one slice per instance $Di[Y,a]$ (e.g., $Quaker(Jane) \supset Pacifist(Jane)$) of that default. [3]

[3]This result depends on the assumption that all functions are fixed (page 36).

Proof : In this case, each open default axiom's default pre-order is equivalent to the conjunction of (i.e., the parallel prioritization of) all of its default instances' default pre-orders. Thus apply Theorem 2.47 and Lemma 3.23 (open default as module of instance defaults) and Corollary 5.28 (canonical decomposition of a module) to decompose within each one-by-one slice that results from Corollary 5.21. □

No Conflict Within an Instance Slice:
This **instance-grain decomposition entirely eliminates the possibility of conflict within a slice** (unless one views the fixture axioms as being polar defaults cf. Lemma 3.42).

Uses of Instance Grain Decomposition in Special Cases:
In chapter 6, we will repeatedly use this instance-grain decomposition, and the instance-grain perspective, while attacking special cases. See sections 6.5 and 6.3.

In addition, our results on the total-propositional special case in sub-section 5.13.2 revolve around instance-grain decomposition. But there, the prioritization is total over the instances, unlike above, where it is parallel, i.e., empty.

5.5 Canonical Serial Decomposition Along Prioritization

Guide to Reader: We defined and motivated serial decomposition and series circumscription in subsection 4.5.4. The gist of the section is the Overview and Observation 5.30.
Overview:
In this section, we show that one can use prioritization as a basis for serial decomposition, by employing the very same canonical protection tactic that we defined (Definition 5.17) in the last section for conjunctive decomposition.

Intuitively, it seems that prioritization as a kind of semantic precedence ought by rights to manifest as a kind of precedence in the process of drawing inferences sequentially. Lifschitz' previous result (Theorem 5.2) for the layered case of predicate circumscription bore this intuition out. In his result, strictly higher prioritization of one layer over another is equivalent to serial precedence in decomposition. Essentially, his result was only about *totally* ordered prioritization: the layers were treated as primitive modules, in effect.

In this section, we show, for the first time, how to extend serial decomposition even to cases with *partially* ordered prioritization: most basically, to the *parallel* case; and more complexly, to any *non-layered* (well-founded) case.

Non-layered prioritization makes matters MUCH more complicated mathematically (just as it did in conjunctive decomposition and in the very definition of prioritization).

We have two main theorems about serial decompositions on the basis of prioritization. The first says that if one applies essentially the same canonical protection tactic as we defined earlier for conjunctive decomposition, in a sequence (e.g., of single-default policies) that is "descending with respect to priority", then the resulting series circumscription is equivalent to the global circumscription. (The constituent axiom sets in the serial case differ from those in the conjunctive case only in their bases: all constituents except for the first contain no base axioms. This difference is only to be expected given the concept of a serial decomposition.) By "descending", we mean that the sequence is topologically sorted with respect to, i.e., "in accordance with descending direction of", the prioritization partial order R. This corresponds to the intuition that higher priority means inferential precedence, which is found in many pieces of work on non-monotonic reasoning (see Observation 5.73 and sections 8.5, 8.7, and 8.8 for more discussion of this relationship). Moreover, one can decompose by modules, *hierarchically*, when the prioritization has been formed by composition, just as in canonical conjunctive decomposition along prioritization (subsection 5.4.4).

Our second theorem we found surprising. It says that, given well-foundedness (Definition 3.18), e.g., quasi-propositionality (Definition 3.22), then the serial decomposition (again, with the canonical protection tactic) may be in *any* sequence, even starting with lowest priority defaults and proceeding in the exact *reverse* of the prioritization partial order. We thus see that canonical protection tactic provides a *robust modularity* in the sense of permitting *sequence-independence* in serial decomposition as well as in conjunctive decomposition.

Observation 5.30 (Additional Results; Guard Reduction)

Next, we give these two theorems for the general case of prioritized circumscription, in guarded form. In addition, the same special cases (e.g., predicates one-by-one, defaults one-by-one, modules), the same methods of guard reduction to ordinary form, and the same ability to hierarchically decompose by exploiting modules and (de)composition of prioritization, right down to finest instance-grain, apply as in canonical conjunctive decomposition cf. the last section. Thus, for example, in CLD, the same constituent axiom sets can be employed (except for their bases, as we mentioned above). For the sake of brevity, however, we omit elaborating these implications in detail.

Definition 5.31 (Descending Priority Sequence)

Let σ be a sequence of indices that are elements of an index set N, over which a prioritization partial order R is defined. We say that the sequence σ is *descending* with respect to prioritization when it is topologically sorted, in the descending direction, according to the prioritization partial order R. So that

$$\forall i, j. \, (i < j) \supset \neg R(\sigma(j), \sigma(i))$$

(Here $<$ stands for ordinary arithmetic less-than. Recall σ notation from Definition 4.12.) That is, no index in the sequence (say, at position j in the sequence) has higher priority than any index that comes earlier in the sequence (say, at position i).

Theorem 5.32 (Serial, Descending, General Priority: Pre-Orders)

Let σ be a permutation of the index set N that is descending with respect to the prioritization partial order R. Let N be finite. Then

$$C(B;(H^N;R);Z) \equiv$$

$$\underset{\substack{i \in N \\ via \\ \sigma}}{///} C(\underset{B}{\bullet};H_i; fix\ H^{R_D(i)}; guard\ (H^{R_I(i)};R^{R_I(i)});Z)$$

Proof Overview: Complicated. By well-founded induction on the prioritization partial order. Uses Theorem 5.14 plus lemmas about properties of prioritized pre-orders. See Appendix. □

Special Case: Predicates, Fine Grain:

$$PPC(B;P^N;R; fix\ W;Z) \equiv$$

$$\underset{\substack{i \in N \\ via \\ \sigma}}{///} PPC(\underset{B}{\bullet};P_i;\emptyset; fix\ P^{R_D(i)},W; guard\ (P^{R_I(i)};R^{R_I(i)});Z)$$

Special Case: Defaults, Fine Grain:

$$PDC(B;D^N;R; fix\ F;Z) \equiv$$

$$\underset{\substack{i \in N \\ via \\ \sigma}}{///} PDC(\underset{B}{\bullet};D_i;\emptyset; fix\ D^{R_D(i)},F; guard\ (D^{R_I(i)};R^{R_I(i)});Z)$$

Incomparability in Prioritization Generates Sequencing Options:
Theorem 5.32 implies that in the case of two parallel minimized pre-orders, it does not matter which phase goes first. More generally, options about sequencings are generated by (prioritization-)incomparability. Moreover, as with conjunctive decomposition in Theorem 5.14, one has a choice of hierarchical scope and grain of the decomposition.

Example 5.33 (Alternative Sequences: Inheritance of Skin Properties)
In the skin-properties inheritance example (Figure 2.4 and Example 5.23), one can canonically serially decompose the global into four phases, one per default, in each of which there is only that one default (i.e., "one-by-one"). The allowed alternative sequences are:

$\langle 1,3,2,4\rangle$, $\langle 1,3,4,2\rangle$, $\langle 3,1,2,4\rangle$, $\langle 3,1,4,2\rangle$

and:

$\langle 1,2,3,4\rangle$, $\langle 3,4,1,2\rangle$

I.e., all sequences are permitted in which the dolphin default's phase precedes the mammal default's phase, and in which the albino-elephant default's phase precedes the elephant default's phase. The mammal default's phase may precede the albino-elephant default's phase, or vice versa, for example.

Or, at a coarser grain size, one can decompose the global into two phases: one per column module. Since these two modules are in parallel with each other, either may go first in the sequence. When further decomposed into one-by-one grain size, these generate the last two of the one-by-one sequences listed above.

Example 5.34 (Descending Sequences: Meetings)
In the Meetings example (Figure 2.1, sections 2.6 and 2.8, Example 5.22), two of the allowed alternative one-by-one sequences are:

$\langle 1, 2, 4, 5, 3, 6 \rangle$

and:

$\langle 1, 5, 4, 6, 2, 3 \rangle$

The latter might be generated, for example, from the coarser-grain sequence of module phases:

$\langle emergencies, attendance, meetings \rangle$

Sequence Independent of Priority:
We find the next result rather suprising. (Moreover, it was very difficult to prove.) Given well-foundedness (Definition 3.18), the phases in Theorem 5.32 may be in *any* sequence. In other words, one can draw conclusions about the lower-priority defaults *before* those for higher-priority defaults, and still get the right answer. Indeed, it is OK for the phase sequence to be the exact **reverse of priority**! We thus see that our canonical protection tactic provides a *robust modularity* in the sense of permitting *sequence-independence* in decomposition, both in conjunctive and in serial decomposition. Note that the well-foundedness condition is satisfied for any propositional theory, or whenever domain closure holds.

Theorem 5.35 (Serial, Any Sequence, General Priority: Pre-Orders)
In Theorem 5.32, suppose that B is well-founded (Definition 3.18) with respect to every serial policy that corresponds to any sequence of the phase policies. (See the proof for the definition of such well-foundedness with respect to serial policies.) Quasi-propositionality (Theorem 3.24) of a global PDC, for example, suffices to ensure such well-foundedness in the prioritized-default case. (Note that the condition is the usual well-foundedness, *not* "well-foundedness".)

Then the sequencing σ may be *any* permutation of the index set N.

Proof Overview: Complicated. A nested induction on permutation of the sequences, using a central intuition of a vacuum cleaner or snowball (seriously!). Uses Theorem 5.32 and lemmas for the 2-ary case. See Appendix. □

Observation 5.36 (Novel Type of Result)
We are unaware of an analogous result, i.e., permitting sequence of serial non-monotonic conclusion-drawing to violate priority sequence, for any other non-monotonic logical formalism.

Example 5.37 (More Sequences: Inheritance of Skin Properties)

In Example 5.33, suppose we add a domain closure axiom (Definition 2.23) to the base. Then one can canonically serially decompose one-by-one in *any* sequence. E.g., the sequence

$\langle 2, 4, 1, 3 \rangle$

is permitted: i.e., mammal then elephant then dolphin then albino-elephant. That is, one may draw NM conclusions about less specific classes' defaults before those about more specific classes' defaults.

Example 5.38 (More Alternative Sequences: Meetings)
Likewise, in Example 5.34, suppose we add a domain closure axiom (Definition 2.23) to the base. Then the module sequence

$\langle attendance, meetings, emergencies \rangle$

is permitted, as is the one-by-one sequence

$\langle 6, 5, 4, 3, 2, 1 \rangle$

5.6 Non-Superfluity of Protection Conditions

At this point, you may be wondering something along the lines of: "These canonical decompositions along prioritization sure are complicated-looking. Isn't there some simpler form? Are all these protection conditions in the decomposition theorems really necessary? Can't one more simply replace guards by fixtures? Does one really have to bother about the guards themselves having prioritization?" In short, you may be wondering whether the protection conditions (and their mathematical complexity) are superfluous.

In this subsection, we will give evidence, mainly by fairly simple and generic CLD counterexamples, that the protection conditions in our (canonical decomposition along prioritization) results are not superfluous, i.e., cannot in general be dropped (or simplified in certain ways).

General Case Means Worst Case:
Remember, firstly, however, that our results are general: they apply to any prioritized circumscription. They thus cover, i.e., have to protect against, the worst-case possibilities of conflicting interaction. In special cases one can often do better, of course. Later, we will give and discuss some special-case results where the protection conditions can be much simplified (e.g., in section 5.13 and chapter 6).

Observation 5.39 (Need to Fix Higher-Priority)
By CLD counterexample. Consider the Pacifism example with priority: Example 1.5. Our canonical protection tactic tells one to fix the Republican default's default formula in the slice or phase for the Quaker default. Suppose, instead, one omits this protection fixture. Then (the guarantee of) equivalence between the "decomposition" and the actual, correct global theory is lost. In particular:

Conjunctive (de)composition: What happens is that the "slice" with the Quaker default generates the conclusion that Nixon is a Pacifist, in contradiction to the correct global conclusion that he is not. In other words, omitting the protection fixture allows the lower-priority default to affect the turf of the higher-priority default, even when the two defaults conflict and the priority should resolve the conflict in favor of the higher-priority default.

Serial (de)composition: The serial case involves a more subtle difficulty. Suppose we extend the Pacifism example a bit further, by adding some more axioms:

$\bullet\!> \; Republican(James)$

$\bullet\!> \; Pacifist(James) \vee Pacifist(Nixon)$

Consider the descending sequence (again, omitting the protection fixture). Then the first (higher-priority, Republican default) phase generates the conclusion that

$\neg Pacifist(James) \vee \neg Pacifist(Nixon)$

and the second (lower-priority, Quaker default) "phase" generates the conclusion that Nixon is a Pacifist. But the correct global theory does **not** include any conclusion either way about Nixon's Pacifism.

Here, in the correct global theory, there is unresolved conflict among the higher-priority defaults, namely between the Nixon instance of the Republican default and the James instance of the Republican default. Since the Quaker default is lower-priority than either of those Republican default instances, it in effect must "stay out of their fight". Intuitively, the higher-priority defaults say "Who needs to listen to you?" to the lower-priority. Omitting the protection fixture allows the lower-priority default to affect the turf of the higher-priority defaults, e.g., the Nixon instance of the Republican default. Intuitively, this contravenes the fact that that turf is the subject of a stalemated battle between greater (higher-priority) powers than the upstart (lower-priority) Quaker default. More precisely, in prioritized circumscription, the lower-priority cannot beat a higher-priority just because that higher-priority is involved in conflict with another higher-priority. Thus the protection fixture, in the serial case, ensures that such **unresolved higher-priority conflicts appropriately remain stalemated**.

Observation 5.40 (Really Only Need to Guard Higher-Priority)

Actually, there is a further subtlety to the question of the need, in general, to fix the higher-priority (defaults' default formulas) in conjunctive decomposition along prioritization. Really, only one side of protection is required: the higher-priority must be at least guarded. Indeed, there is an equivalent, alternative form of the canonical protection tactic: in which the higher-priority (e.g., $H^{R_D(i)}$ in Theorem 5.14) are guarded rather than fixed. However, this alternative form, where some fixing is replaced by guarding, does not appear to particularly simplify the protection conditions (e.g., expressively). Thus, for the sake of focus, we omit further discussion about this alternative form of the canonical protection tactic.

Observation 5.41 (Need to Guard Incomparable-Priority)

By CLD counterexample, for the most basic case: parallel (empty) global priority. Consider the Pacifism example without priority: Example 1.4. Our canonical protection tactic tells one to guard the Republican default's default formula in the slice or phase for the Quaker default, and vice versa.

Suppose, instead, one omits this protection guard. Then (the guarantee of) equivalence between the "decomposition" and the actual, correct global theory is lost. In particular:

Conjunctive (de)composition: What happens is that the "slice" with the Quaker default generates the conclusion that Nixon is a Pacifist, and the "slice" with the Republican default generates the conclusion that he is not. Conjoining these two "slices" results in contradiction. This is precisely the situation illustrated in Figure 4.1 (sub-section 4.3.2) that we used to exemplify logical non-modularity. By contrast, in the correct global theory, no conclusion about Nixon's Pacifism is entailed. In other words, omitting the protection guard allows a default to increase at the expense of another, incomparable-priority default, even though this domination is not legitimized by any strict priority.

Serial (de)composition: Consider the sequence in which the Quaker default's "phase" comes first. What happens is that this "phase" generates the conclusion that Nixon is a Pacifist. By contrast, in the correct global theory, no conclusion about Nixon's Pacifism is entailed. Likewise, in the sequence in which the Republican default's "phase" comes first, that "phase" generates the conclusion that Nixon is not a Pacifist. Thus, for either sequence, the serial "decomposition" is incorrect. Again, omitting the protection guard allows a default (in the serial case, whose "phase" comes earlier in the sequence) to increase at the expense of another, incomparable-priority default, even though this domination is not legitimated by any strict priority.

Observation 5.42 (Cannot Just Fix Incomparable-Priority)
Example 5.4 showed, by counterexample, that fixing the incomparable-priority (defaults' default formulas) instead of guarding them does not guarantee equivalence of the "decomposition" to the global, even in case of globally parallel priority. In prioritized circumscription, incomparable-priority permits **co-operative agreement** in a particular sense. E.g., two defaults may both lead to the same non-monotonic conclusion: but if each (more precisely, each's default formula) is paralyzed by fixture when the other is "operating", then the "co-operative" conclusion is prevented.

We illustrate this point further with a variant of the Pacifism example without priorities. Extend Example 1.4 to include the additional axioms:

- •> $Quaker(Mary)$
- •> $Republican(Joe)$
- •> $Pacifist(Mary) \equiv \neg Pacifist(Joe)$

(say, Mary and Joe are friends but disagree on all political matters). In the correct global theory, the Quaker default for instance Mary generates the conclusion that Mary is a Pacifist. The Republican default for instance Joe generates the conclusion that Joe is not a Pacifist. The two defaults in effect co-operatively agree on these default conclusions, whose truth value is forced to be equivalent by the for-sure information.

Suppose one fixes, instead of guards, the incomparable-priority. Then:

Conjunctive (de)composition: The "slice" with the Quaker default cannot generate the conclusion that Mary is a Pacifist. Likewise, the "slice" with the Republican default cannot generate the conclusion that Joe is not a Pacifist. Their conjunction cannot generate either of these conclusions, and is thus not equivalent to the correct global theory.

Serial (de)composition: Ditto for the "phases", in either sequence. Thus the final tier is not equivalent to the correct global theory.

Observation 5.43 (Need Prioritization Within Guard)

Perhaps the mathematically hairiest aspect of the canonical protection tactic is the prioritization *within* the guard. Next, we show why this is, in general, necessary.

By CLD counterexample. We begin by giving an example in abstract, mathematical form. Afterwards, we discuss how such an example might arise in applications. In this example, there are four defaults $\langle D1, D2, D3, D4\rangle$ (open, and, for simplicity, unary), whose prioritization R has the form of two columns in parallel. I.e., in this respect the situation is isomorphic to Figure 2.4, or the top of Figure 5.25.

$$(d1) :> \; D1[Z, x]$$
$$(d2) :> \; D2[Z, x]$$
$$(d3) :> \; D3[Z, x]$$
$$(d4) :> \; D4[Z, x]$$
$$\mathcal{PREFER}(d1, d2)$$
$$\mathcal{PREFER}(d3, d4)$$

In addition, the base B includes that: $D1$ is equivalent to $D3$ and that $D2$ is equivalent to $D4$:

$$\bullet\!> \; \forall x.\, D1[Z, x] \equiv D3[Z, x]$$
$$\bullet\!> \; \forall x.\, D2[Z, x] \equiv D4[Z, x]$$

; and that there is conflict between the default $D1$ and the default $D2$ for an individual a:

$$\bullet\!> \; \neg(D1[Z, a] \wedge D2[Z, a])$$

Then in the correct global theory, (the default) $D1$ prevails in the conflict with $D2$ about a, since it has higher-priority:

$$\models\!\!\approx \; D1[Z, a]$$
$$\models\!\!\approx \; \neg D2[Z, a]$$

Notice also that, due to the equivalences:

$$\models\!\!\approx \; D3[Z, a]$$
$$\models\!\!\approx \; \neg D4[Z, a]$$

That is, $D3$ and $D4$ also conflict about a, and $D3$ prevails, since it has higher-priority.

Consider decomposing this axiom set one-by-one. Our canonical protection tactic tells one to guard $D3$ and $D4$ in the slice or phase where the default $D1$ is maximized. Moreover, it tells one to prioritize within the guard: to make $D3$ have higher priority than $D4$ within the guard pre-order. Likewise, it tells one to employ a prioritized column pre-order guard in each of the other slices or phases. Suppose, instead, one omits the prioritization within the protection guard. E.g., in the $D1$-maximizing slice, suppose one guards $D3$ and $D4$ in parallel. Then (the guarantee of) equivalence between the "decomposition" and the actual, correct global theory is lost. In particular:

Conjunctive (de)composition: What happens is that the "slice" with default $D1$ cannot generate the correct conclusion $D1[Z, a]$. I.e., the default does not go through, on instance a, as it should. It is prevented from doing so by the guard on $D4$. Remember, (the base implies that) the formula $D4$ is equivalent to the formula $D2$. Thus $D1$ increasing, as the maximization of $D1$ is attempting, must come at the expense of $D4$ decreasing, which is prohibited by the guard on $D4$. By contrast, when the guard is prioritized (i.e., in the canonical protection tactic), then $D4$ is permitted to decrease as long as $D3$ (which here is equivalent to $D1$) is increased. Likewise, the "slice" with default $D3$ cannot generate the correct conclusion $D3[Z, a]$ (which is equivalent to $D1[Z, a]$); it is prevented from doing so by the guard on $D2$. (And neither the "slice" with default $D2$ nor the "slice" with default $D4$ can remedy matters.) In short: **omitting the prioritization within the guarding of the incomparable-priority makes that guarding too strong a constraint** (see also Corollary 5.45 below). In this example, some of the defaults that have incomparable priority with respect to each other need to co-operate in a way that nevertheless involves prioritization dominance with respect to yet other (i.e., third-party and fourth-party) defaults. That is, the need for co-operation between incomparable-priority defaults, that we saw in the case of globally parallel priority in Observation 5.42, extends to **co-operating to resolve conflict (with yet other defaults) via priority dominance**.

Serial (de)composition: Ditto for the "phases", in any sequence. Thus the final tier is not equivalent to the correct global theory.

A more intuitively meaningful counterexample: The following is a similar, slightly more complex, counterexample.

Example 5.44 (Overlapping Pacifism)
We extend the Pacifism example with priorities (Example 1.5) to be the global CLD axiom set that contains exactly:

$(d1) :> \ Republican(x) \supset \neg Pacifist(x)$

$(d2) :> \ Quaker(x) \supset Pacifist(x)$

$\mathcal{PREFER}(d1, d2)$

$\bullet\!> \ \forall x.\, Resides(x, Baketown) \wedge Republican(x)$
$\qquad\qquad \supset \ Politically\_Conservative(x)$

$\bullet\!> \ \forall x.\, Resides(x, Baketown) \wedge Religiously\_Non\_Violent(x) \ \supset \ Quaker(x)$

$(d3) :> \ Politically\_Conservative(x) \supset \neg Pacifist(x)$

$(d4) :> \ Religiously\_Non\_Violent(x) \supset Pacifist(x)$

$\mathcal{PREFER}(d3, d4)$

$\bullet\!> \ Resides(Sue, Baketown)$

$\bullet\!> \ Republican(Sue)$

$\bullet\!> \ Religiously\_Non\_Violent(Sue)$

$\bullet\!> \ Resides(Jeb, Baketown)$

$\bullet\!> \ Quaker(Jeb)$

$\bullet\!> \ Politically\_Conservative(Jeb)$

- •> $Resides(Fred, Baketown)$
- •> $Republican(Fred)$
- •> $Politically\_Conservative(Ali)$
- •> $Religiously\_Non\_Violent(Teresa)$
- •> $Quaker(Jane)$

The correct global theory includes the conclusions that Sue and Jeb are not Pacifists. But, as in the last counterexample (Observation 5.43), without the prioritization within the guards, these conclusions are not generated.

Corollary 5.45 (Increasing Prioritization Weakens a Guard)
Increasing the prioritization[4] within a guard pre-order weakens the effect of that guard: the guarded circumscription is stronger; the guard is less of a constraint on the minimization. More precisely:

$$[\forall xy.\ R1(x,y) \supset R2(x,y)] \quad \Rightarrow$$

$$[C(B; H; guard\,(G; R2); Z) \ \models\ C(B; H; guard\,(G; R1); Z)]$$

where $R2$ is defined over the same indices (e.g., N) as $R1$.

Proof : Modify slightly the argument in the proof of Theorem 2.58. □

Perspective: Relationship to Monotonicity of Prioritization:
One can view the monotonicity of prioritization (Theorem 2.58) in terms of its effect on the guards in the canonical conjunctive decomposition along prioritization cf. Theorem 5.14 (also, the guarded forms of Corollaries 5.18, 5.21,5.28). As the global prioritization is increased, there is (non-strictly) increased prioritization within the guard pre-order in each of the slices the canonical decomposition along prioritization. (Here, we are referring to the guarding of the incomparable-priority.)

Observation 5.46 (Layered Means No Prioritization Within Guard)
Recall by property 3. of Observation 2.50 that: If the global prioritization is layered, then the prioritization within the incomparable-priority is always empty. Thus in the globally layered case, there is no need to prioritize within the guarding arising from canonical decomposition along prioritization. This gives some perspective on just how the layered case is simpler, at least mathematically, than the non-layered case. However, we have found that many, if not most applications of interest, e.g., for learning agents and embedding inheritance, are non-layered (see discussion in chapters 1 and 2).

Observation 5.47 (Ubiquity of Need for Protection Conditions)

[4] i.e., adding more pairs (paths) to the prioritization partial order relation (dag)

How common is the need for the protection conditions in decomposing (conjunctively or serially) the kinds of CLD axiom sets that are likely to arise in applications? We argue that the need is ubiquitous. The whole point of default reasoning is to allow for violation (of defaults) and conflict (between defaults). Whenever there is conflict between defaults that is *not* resolved by (dominating) priority, there is a need to guard the incomparable-priority. Whenever there is conflict between defaults that *is* resolved by (dominating) priority, there is a need to fix (more precisely, to guard; see Observation 5.40) the higher-priority. Thus **whenever there is conflict**, there is a need at least to guard with respect to the higher-priority and the incomparable-priority.

The need for prioritization within guarding is a more subtle and open question. The last counterexample we gave in Observation 5.43 involves a kind of equivalence of defaults. Such redundancy, however, need only be partial, and relative to a particular set of for-sure axioms. We believe, therefore, that the existence of such interaction effects will be common in large, expressively rich non-monotonic axiom sets. For example, in current large, expressively rich first-order monotonic axiom sets, redundancy and definitional equivalence are very common.

5.7 Reducibility of Prioritization

Observation 5.48 (Prioritization is Reducible)
Perhaps the most interesting, fundamental pay-off of our results about canonical decomposition along prioritization is that they show how to **expressively reduce prioritization**: to eliminate it completely, in a sense; or, more generally, to simplify it (in case of hierarchical decomposition). Formally, we can view canonical conjunctive (or serial) decomposition along prioritization as an equivalence-preserving **reformulation transformation**. It takes a global axiom set or circumscription and maps it into a conjunctive (or serial) collection of axiom sets or circumscriptions. (More precisely, this equivalence is up to conservative extension, i.e, with respect to the GPT-free sub-language, where GPT are the placeholder, newly-introduced, guard predicates.)

Thus the expressive class of prioritized default circumscriptions is reducible to the expressive class of conjunctions, or series, of un-prioritized (i.e., parallel) single-default circumscriptions.

Ditto for predicate case: the class of prioritized predicate circumscriptions is reducible to the class of conjunctions, or series, of single-predicate circumscriptions.

Of course, in any kind of expressive reduction, there are several issues to consider in assessing its significance. One is whether there is some nasty kind of blow-up in the size of the representation when it is expressively reduced. Another is how much computational work is involved in the reformulation transformation. Related to this is how "local" or "modular" the transformation between representations is.[5] For example, first-order (or propositional) circumscription often is (and skeptical Default Logic always is) equivalent to a formula in first-order (or propositional) logic. Thus, in such a case, it is expressively reducible. But there can be lots of computational work and expressive blowup involved in this "transformation".

[5] e.g., "modular" in the sense discussed in [Imielinski, 1987]

In section 5.11 (at the end), we address: the expressive complexity of the transformed representations that result from canonical decomposition; and we address how much compuational effort is involved in performing the transformation. (Also see chapter 7, especially sub-section 7.7.5, for more about computational issues involved in exploiting canonical decomposition.) We show that the degree of blowup and amount of computational effort is far less than is involved in, say, (in general in) reducing prioritized default circumscription to first-order (recall the discussion of reducibility in sub-section 4.3.1).

5.8 Canonical Derivability

In this section, we show how to define a canonical concept of derivability from a subset of the starting pre-orders, e.g., derivability from a subset of the global defaults. The "obvious" **simplistic approach** is to say a conclusion is derived from a subset of defaults if those defaults (and their prioritization) plus the base entail the conclusion. However, this **is unsound**, in general, because it may be that some of the rest of the global set of defaults conflict with some of the defaults in this subset. We use, instead, the conjunctive combination of the canonical slices that correspond to each default in the subset. This ensures soundness, as well as completeness in the sense discussed in Observation 4.3.

Our motivation in defining such a concept of derivability is twofold. Researchers in and out of the NMR community very often speak intuitively of deriving a default conclusion from an individual default: e.g., deriving the conclusion that Nixon is not a Pacifist from the default that Republicans are not Pacifists. Thus, firstly, it is interesting conceptually to provide a rigorous basis for this intuitive, **informal usage**. Secondly, a concept of derivability is interesting practically and theoretically as a basis for **selectiveness in inference** (selectiveness in the sense discussed in Observation 4.4), and for the development of more general **proof theory** for prioritized circumscription than is currently available. In particular, it is a step towards the development of proof theory for predicate and default circumscriptions with **non-layered** priority, for which there is no previous proof theory available at all.

Next, we use canonical conjunctive decomposition to define a canonical notion of derivability. There are alternative ways, independent of decomposition, to define such a notion, but we will not take the space here to detail them.

Definition 5.49 (Derived Solely From a Subset of Defaults)
Relative to a prioritized default circumscription $PDC(B; D^N; R; fix\ F; Z)$, or, more generally, relative to a guarded prioritized circumscription
$PDC(B; D^N; R; fix\ F; guard\ G; Z)$:
Let $S \subseteq N$ index a subset of the defaults. We say that a conclusion $E[Z]$ is *derived solely from* that subset of the defaults when $E[Z]$ follows from the conjunction of the *slices* corresponding to S (i.e., whose maximized default's index is in S), in the canonical conjunctive decomposition cf. Corollary 5.21 I) and III) (first equivalence form in the latter).

Definition 5.50 (Derived Solely, General)

Relative to a prioritized circumscription $C(B;(H^N;R);Z)$, or, more generally, relative to a guarded prioritized circumscription $C(B;(H^N;R);guard\ G;Z)$:

Let $S \subseteq N$ index a subset of the starting pre-orders (e.g., corresponding to modules). We say that a conclusion $E[Z]$ is *derived solely from* that subset of pre-orders (e.g., predicates or modules) when $E[Z]$ follows from the conjunction of the *slices* corresponding to S (i.e., whose minimized pre-order's index is in S), in the canonical conjunctive decomposition cf. Theorems 5.14 and 5.15 (first equivalence form in the latter).

Definition 5.51 (Justifying Set of Defaults)

In Definition 5.49, we say that S is (i.e., indexes) a *justifying set of defaults* for the conclusion E when S is a *minimal* set of defaults such that E can be derived solely from S. By minimal here, we mean that E cannot be derived solely from any proper subset of S.

Note that, in general, a given conclusion may have more than one justifying set of defaults.

The notion of a justifying set of defaults formalizes the idea that a NM conclusion may be based on several defaults taken together, but on no smaller subset (e.g., no individual one) of those defaults.

Our concept of a justifying set of defaults is somewhat similar to Reiter's [1980] concept of a "generating set of defaults" in Default Logic. There are, however, two important differences. The first is that our concept is relative to a skeptical NM formalism, whereas Reiter's is relative to a credulous NM formalism (recall discussion of skeptical versus credulous on page 23). The second difference is that our concept handles prioritization, whereas Reiter's does not (Default Logic lacks priorities). See section 8.5 for more discussion of Default Logic and its relationship to circumscription. Using Observation 8.1 there, our concept of a justifying set of defaults can be applied to the skeptical version (which we define there) of Default Logic.

Example 5.52 (Justifying Sets of Defaults, Pacifists)

Consider Example 5.44 (Overlapping Pacifists), extended by adding the following axioms.

$(d5) :> \ Pacifist(x) \supset Against(x, US\_in\_Vietnam\_War)$

$(d6) :> \neg Pacifist(x) \supset Against(x, Eliminate\_Dept\_of\_Defense)$

Then, for the NM conclusion

$Against(Teresa, US\_in\_Vietnam\_War)$

there is a unique justifying set of defaults:

$\langle d4, d5 \rangle$

By contrast, for the NM conclusion

$Against(Sue, Eliminate\_Dept\_of\_Defense)$

there are two different justifying sets of defaults,

$\langle d1, d6 \rangle \qquad \langle d3, d6 \rangle$

since her non-Pacifism may be concluded either from the Republican default or from the Politically-Conservative default.

Use For Safety of Updating:
In section 5.12, we will use canonical derivability to formulate and prove a fundamental result about prioritization as a basis for safety of updating with new defaults.

5.9 Implications for Computing Inferences

Our canonical decompositions along prioritization have many implications for computing inferences in prioritized circumscriptions, and for relationships to other NM formalisms.

Selectiveness in forward inference: (as discussed in Observation 4.4) Exhaustive forward inference on a conjunction of one or more slices generates only a part of the global NM theory. Similarly, exhaustive forward inference on a tier (series of phases) generates only a part of the global NM theory.

Locality in inference: In inference, one need only work with (an axiom set containing) one default (axiom) at a time (see also Theorem 5.29); or, more generally, only with a subset of the defaults at a time.

Concurrency in inference: (as discussed in Observation 4.5) NM inference in different slices can be performed concurrently; the combination only involves monotonic inference. NM inference can be serialized (into phases) in many different sequencings.

Steps towards fuller proof theory: We invented non-layered prioritization in circumscription. There is no previous proof theory available for even the propositional-predicate case of circumscription with non-layered priorities. Our canonical decomposition results and the notion of canonical derivability take a signficant step towards this goal: they reduce the problem to that of inference in conjunctions of, or series of, parallel circumscriptions. Proof theory and inference methods for parallel circumscription, which are available, can then be applied. In particular, our decompositions enable:

Hand-computing results of non-layered circumscriptions.

Direct use of guarded forms: Note that it is often possible to work directly with guarded circumscriptions to show NM entailments: by comparing models, especially for small propositional examples. Note that guarded forms arising from canonical decomposition obey most of the well-behavior properties associated with various expressive clases of the originating ordinary global circumscriptions. In particular, our results about well-foundedness, "well-foundedness", satisfiability, and immunity of the fixed, from section 3.4 and subsection 3.5.3, all generalize straightforwardly to the guarded forms of canonical slices and phases. Note also that our results on conservative extension for abnormality theories and fixtures, in section 3.3 and 3.5, also generalize straightforwardly: a guard can be present.

We will return to these issues later: in chapters 7 and 8.

5.10 Analogy to Programs with Side Effects

We remark that our results about canonical decompositions along prioritization suggest an interesting analogy between CLD axiom sets, and programs in programming languages with side effects.

The constituent CLD axiom sets (e.g., corresponding to a module) in a decomposition of a given global CLD axiom set are analogous to the sub-programs of a given global program.

In our NM decompositions, the role of a non-local variable in a programming language is played by the truth value of a non-local default's formula. (By non-local default, here, we mean a default axiom that is present in the global axiom set but that is not present as a default axiom in the given constituent axiom set under consideration.) That is, in the case of NMR, the side-effects are the truth values of sentences that are the subject of potential conflict between that constituent axiom set's (e.g., a module's) entailment ("execution") and the rest of the global axiom set.

In programming languages with side effects, one summarizes relevant external context with declarations about non-locals read and written. In our NM decompositions, the protection-condition axioms play the role of these declarations.

5.11 Expressive Complexity

One of our main motivations for developing the ideas of conjunctive and serial decomposition, and for investigating them for prioritized circumscription, is that we hope to be able to do inference in, and to organize safety in belief revision in terms of, simpler and more local sub-theories (see section 4.5).

The good news is that each slice or phase in our canonical decompositions along prioritization is guaranteed to be simpler than the global in one very important respect: the set of defaults is always smaller (smaller in the sense of being a proper subset). Indeed, one can always arrange the grain size of the decomposition to make the set of defaults in each slice or phase be just a singleton. In sections 4.3 and 4.4, we argued that much of the difficulty of default reasoning is due to the potential for conflict between defaults, not just between defaults and the for-sure axioms. Thus our decomposition results appear to be a significant step towards achieving locality, and cutting down the potential combinatorics involved in conflict.

A difficulty with the general case of our canonical decomposition results, however, is that the slices or phases are not always clearly simpler than the global circumscription. By the above **criterion of fewer defaults** (and simpler prioritization), they always are. But there are other possible criteria: other dimensions of the expressive complexity of the slices or phases include the base and fixture axioms.

For example, each slice (or phase) contains at least as many base axioms as the global. And the additional base axioms in each slice or phase may involve an increase in expressive complexity relative to the global base, in a way that goes beyond just an increase in size. In particular: even when the global base and default formulas are universal (e.g., clausal), the base axioms introduced during guard reduction may involve existential quantifiers, when the guards being reduced themselves have prioritization.

Furthermore, if one views the introduced fixtures as pairs of polar defaults cf. Lemma 3.42, then each slice or phase contains at least as many defaults as the global. Intuitively, it seems to us that a fixture is usually simpler than a default. However, it is currently unclear how fixtures affect the computational complexity of inference, both in current inference procedures for circumscription, and intrinsically.

Nevertheless, some slices or phases may be indeed clearly simpler than the global, even in a decomposition where some, or most, slices or phases are not. For example, in the Meetings example (Figure 2.1 and section 2.8), the slice for the "emergencies" module is *clean* (Definition 4.7) and is clearly simpler than the global.

Next, we give some more detailed analysis of the expressive complexity of canonical conjunctive decomposition along prioritization in CLD, concentrating mainly on how the complexity of the decomposition grows, **worst-case**, in the size of the CLD axiom set. (The analysis for serial is similar, except for the complication that the base in a given phase includes the conclusions from the previous phases' NMR. We omit it for reasons of space.) Remember that in CLD, fixture axioms are treated as distinct from default axioms.

Fewer Defaults; Simpler Prioritization:
The set of defaults in a slice is always a subset of the global: let S be its index. The attendant prioritization in the slice is, likewise, essentially a subset of the global. More precisely, it is a projection of the global R onto the sub-domain S.

In the one-by-one decomposition, there is exactly one default axiom, and no prioritization at all, in each slice. In the instance-grain decomposition (Theorem 5.29), if it is available, there is exactly one **closed** default axiom, and no prioritization at all, in each slice. This **instance-grain decomposition entirely eliminates the possibility of conflict within a slice** (unless one views the fixture axioms as being polar defaults cf. Lemma 3.42).

Fixtures:
In each slice, there is one fixture for each higher-priority external global default's default formula. In addition, there is one fixture for each introduced guard predicate (these arise, remember, from the guard reduction step), if (as is usually the case in applications) there are any incomparable-priority external global defaults. The number of guard predicates is no more than the number of predicates in the original language, i.e., no more than the size of Y. In addition, each slice contains all of the global fixture axioms.

Note that all *formula* fixtures (e.g., the higher-priority defaults' default formulas) can straightforwardly be expressed equivalently via *predicate* fixtures. (Just as all formula defaults can straightforwardly be expressed equivalently via minimized predicate defaults, i.e., abnormalities.) For each formula fixture, a new predicate symbol is introduced, along with a new base axiom stating that the fixed formula is (universally) equivalent to the new predicate. (See Proposition 3.34.) This re-expression involves little increase in the size or syntactic-form complexity of the CLD axiom set if equivalence is permitted freely as a logical connective. However, in clausal form, this may involve, in the worst case, a clausal conversion step[6] to express one side of the introduced definitional-equivalence base axiom. If, as is common, however, the length of any given fixture formula or default formula is bounded in a constant way relative to the size of the global CLD axiom set, then from point of view of size analysis in terms of that global axiom set, this combinatoricity arising from the definitional equivalence need not be too worrisome.

Let d be the size of the syntactic representation of the default axioms in the global axiom set.
Let b be the size of the syntactic representation of the base axioms in the global axiom set.
Let f be the size of the syntactic representation of the fixture axioms in the global axiom set.
Let nd be the number of global defaults.
Let nf be the number of global fixtures.
Let r be the size of the global prioritization partial order. $r = n(n-1)/2$.
Let y be the size of the original (global axiom set's) base language Y. Without loss of generality: y must be less than $b + d + f$. Indeed, it is reasonable to assume that y is less than $b + d$.
Let h_i be the number of higher-priority external global defaults, for slice i.
Let p_i be the number of incomparable-priority external global defaults, for slice i.
Let rp_i be the size of the prioritization partial order among the incomparable-priority external global defaults, for slice i. Note that $rp_i = p_i(p_i - 1)/2$.
Let s_i be the number of default axioms in the slice i (1 for one-by-one, always less then d for hierarchical, module slices).

Thus:

1. The syntactic form of the fixture axioms in each slice (say, slice i) CLD axiom set is no greater in complexity than that of the global CLD axiom set's defaults. E.g., the slice fixture axioms are clausal if the global's are.

2. The size of synactic representation of the slice CLD axiom set's fixture axioms is no greater than $f + d + y$. Typically in the applications we have looked at, this size is dominated by d. The number of the slice's fixture axioms is $f + h_i + y$.

And if the number of higher-priority defaults is relatively small, then the increase in the number of fixtures will not be too much of a problem.

Fixtures Viewed As Pairs of Polar Defaults:
Suppose, however, one views each fixture axiom as a pair of polar default axioms. Then the number of global defaults is $nd + 2nf$. And the number of the slice's default axioms is $s_i + 2nf + 2h_i + 2y$, which is less than $2(nd + 2nf) + 2y$. If, as is common in the applications we have considered, nd dominates nf and y, then the number of slice defaults is less than twice the number of global

[6] Conversion to clausal form is well known to be potentially exponential in space and time.

defaults. Moreover, if the number (h_i) of higher-priority external global defaults is less than the number (p_i) of incomparable-priority external global defaults, as is common for most defaults (i's in one-by-one) in the applications we have considered, then the number of slice defaults is less than the number of global defaults. A further note: Typically, in the applications we have considered, the global fixtures are less complex in syntactic form than the global defaults. Thus the syntactic form of the slice's defaults is no greater in complexity than that of the global defaults. E.g., the slice is clausal if the global is.

More, and Potentially More Expressive, Base Axioms:
Each slice contains all of the global base axioms. In addition, each slice contains base axioms arising from guard reduction, if (as is usual) there are any incomparable-priority external global defaults. Both the size and the syntactic form of these base axioms arising from guard reduction are thorny points, unfortunately. Let us call these axioms: BGR. The difficulties with BGR arise primarily from the potential (and common, in applications) prioritization within the guarding, i.e., among the incomparable-priority defaults. If, however, for a given slice, the incomparable-priority defaults have empty prioritization among themselves, then that slice's base must be at most only moderately more expressively complex, both in size or syntactic form, than the union of the base and default formulas in the global axiom set.

Recall that in Corollary 5.21:

$$BGR \stackrel{\text{def}}{\equiv} (D^{R_I(i)}[GPTi] \preceq_{R^{R_I(i)}} D^{R_I(i)}[Y])$$

Examples 5.23 and 5.19 illustrated what BGR looked like, for various slices, in terms of the Skin-Properties and Meetings examples.

The size difficulty with BGR is that (even permitting equivalence freely as a logical connective, in the usual first-order logical syntax) it may, in the worst case (e.g., when $R^{R_I(i)}$ is a total order), involve up to approximately rp_i (the size of $R^{R_I(i)}$) sub-expressions of the form:

$$\forall xj.\ Dj[GPTi, xj] \equiv Dj[Y, xj]$$

(Let us call these sub-expression $BGRSE$.) But rp_i, remember, is proportional to the square of the number p_i of incomparable-priority external global defaults. And, if as is common in the applications we have looked at, for most slices p_i grows linearly with the number of global defaults nd, then BGR may grow as the **square** of the number of global defaults.

The syntactic-form difficulty with BGR is that these last subexpressions $BGRSE$ also appear on the left-hand-side of implications, i.e., they appear negatively. But $BGRSE$ may contain universal quantifiers ($\forall xj.$ above): for any Dj that is an open, rather than closed, default. Thus, BGR may contain **existential** first-order quantifiers, even when the global base, default, and fixture axioms are all universal. In the applications we have looked at, this situation indeed arises commonly.

Thus, for any slice i in which the prioritization $R^{R_I(i)}$ among the incomparable-priority is non-empty, and in which $R_I(i)$ contains an open default that has higher priority than some other default within $R_I(i)$, then BGR (for that i) will involve an existential quantifier.

Size of Language:

In each slice, the base language is augmented by introduced guard predicates, if (as is usual) there is need for guard reduction. The size of the base language $\mathcal{L}$ employed in each slice is at most $2y$. I.e., it is at most double that of the global: the number of guard predicates is no more than y. Considering all the nd slices, in a one-by-one decomposition: the number of guard predicates introduced overall is no more than nd times y.

Computational Complexity of the Decomposition Transformation:
Given a global CLD axiom set:
The amount of computational effort required to generate all the constituent axiom sets for the canonical conjunctive or serial decomposition is at most $O(a^3)$, where a is the overall size of the global axiom set. The size of all the constituent axiom sets, taken together, is no more than $O(a^3)$. (In standard, i.e., unrestricted, first-order predicate calculus syntax.)

Note that this is **polynomial in the axioms**. By contrast, the set of conclusions, i.e., the NM theory, may be much larger than the set of axioms. This size of the NM theory is well-defined, for example, in case of quasi-propositionality (by Theorem 3.24).

Due to space, and to preserve focus, we omit: our algorithm for computing the constituents; and a more detailed, tighter computational complexity analysis. Both can be extracted relatively straightforwardly from the earlier parts of the section.

See chapter 7 (e.g., sub-section 7.7.5) for more discussion of computational issues revolving around exploiting our canonical decomposition results.

5.12 Safety of Higher-Priority Conclusions

In this section, we show how our canonical decomposition results yield safeties of updating. We use the notion of canonical derivability to show a general result about the safety of higher-priority conclusions, i.e., the safety of all previous conclusions derived solely from the subset of the defaults that are higher-priority than some new update defaults. Again, the "obvious" **simplistic approach fails**: non-layered prioritization makes things much trickier than in the layered case from which you might draw your intuitions. We use canonical derivability to define just what are those higher-priority conclusions.

We begin by formalizing the notion of updating with a single new default.

Definition 5.53 (Default Update)
We define a *default update* to be updating a prioritized default pre-order (typically, the global pre-order for a PDC) with a new default pre-order plus (possibly empty) prioritization of that new default relative to the previous defaults. In CLD, a default update is thus defined to be an update set of axioms containing exactly: one new default axiom plus zero or more prioritization axioms that are only about that new default.
Terminology: By "only about" here, we mean that each prioritization axiom in the update must mention the label of the new default.

We will speak, as well, of default updates to PDC's and PPC's.

Advantage of PDC and CLD Versus Predicate Circumscription:
Notice how much more awkward the concept of a default update is in abnormality-predicate circumscription, which is, in current practice, the most commonly-applied style of predicate circumscription. In abnormality style, one must always talk about an associated base update, and about conclusions mentioning abnormality predicates which are typically intended only as placeholders rather than being of direct interest.

It is also interesting to consider updating, all at once, with a bunch of new defaults that make up a module.

Definition 5.54 (Module Update)
We define a *module update* to be updating a prioritized default pre-order (typically, the global pre-order for a PDC) with a new module pre-order plus (possibly empty) prioritization of that new module relative to the previous defaults, e.g., previous modules. A default update is the special case of a module update where the new module pre-order is a single default pre-order.

In the context of CLD, recall that a module is specified by a collection of default axioms and prioritization axioms only about the prioritization of those default axioms relative to each other. We call that prioritization the *internal prioritization* of the module, in contrast to its *external prioritization* relative to the other pre-orders in the overall global prioritized pre-order. A module update in CLD thus specifies some default axioms, their internal prioritization, and their external prioritization. Note that not every collection of default axioms and prioritization axioms constitutes a module update. In a module update, all the default axioms in the module have the *same* external prioritization relative to the other default axioms not in the module.

In more formal detail:
Let $(H0; R0)$ be a previous global pre-order: a prioritized pre-order with index set N, where each previous starting pre-order in $H0$ is a default pre-order or, a bit more generally, a module pre-order. A module update is specified by the new module pre-order U plus the new prioritization information SU that is only about that new module.
SU is required to be an acyclic binary relation that specifies prioritization information that is only about u, the index of U: each pair of SU has either the form $\langle i, u\rangle$ or the form $\langle u, i\rangle$, where $i \in N$. SU may be empty, however.

Then $H1 \stackrel{\text{def}}{=} H0 \cup \{U\}$ is the new tuple of starting pre-orders; it has index tuple $N1 \stackrel{\text{def}}{=} N \cup \{u\}$. And the new prioritization partial order $R1$, defined over $N1$, is defined to be the transitive closure of: $R0 \cup SU$. ($R0$ and $R1$ here are viewed as binary relations.)

More briefly, we say that $(H1; R1)$ is the result of a module update $\langle U, SU\rangle$ to $(H0; R0)$.

Intuitions about Prioritization and Default Updates:
Consider your intuitions about how the prioritization of the previous defaults relative to a new update default relates to which previous conclusions can be guaranteed to be safe under that default update. Let us begin by considering a scenario in which there is exactly one previous

default. Clearly, one must expect that a conclusion derived from a previous default that has *incomparable* priority relative to the new default may be affected: the new default may introduce conflict with that previous default, and that conflict will not be resolved by strict priority. For example, in the Pacifism example without priorities (Example 1.4), updating with the new Quaker default resulted in the retraction of the previous conclusion that Nixon was not a Pacifist, which was derived from the previous Republican default.

Equally clearly, if a previous conclusion was derived from a default that was *lower* priority than the new default, one must expect that that conclusion is also at risk: the new default may introduce conflict with that previous default, and, if so, that conflict will be resolved by strict priority in favor of the new default.

Also clear, however, is that if a previous conclusion was derived from a default that was *higher* priority than the new default, then one can expect that conclusion to be safe: any conflict with the new default will be resolved by priority in favor of the previous default.

More generally, of course, a given previous conclusion may be based on several previous defaults taken together, but on no smaller subset (e.g., no individual one) of those defaults. These justifying defaults may have different prioritizations relative to the new update default (and relative to each other): some may be higher in priority than the new default, some incomparable, and some lower. Clearly, if *all* of these defaults are higher priority than the new, then one can expect that conclusion to be safe. Otherwise, however, it may be that one of these previous justifying defaults conflicts with the new, but that that previous default is not higher priority than the new. If so, that conflict will not be resolved in the previous' favor by priority. The justification will then be undercut.

Our next result brings this intuition to fruit.

Theorem 5.55 (Safety of Higher-Priority Conclusions)
In PDC and CLD: A default update is partially monotonic with respect to all conclusions derived solely from the previous higher-priority defaults. More precisely: none of the previous conclusions need to be retracted that were derived solely (Definition 5.50) from the previous defaults that are higher-priority than the new default according to the prioritization after the update.

More generally, a module update is partially monotonic with respect to all conclusions derived solely from the previous higher-priority modules (e.g., defaults).

Proof Overview: Non-trivial. Uses canonical conjunctive decomposition: Theorem 5.14. The essence is to use Observation 4.6: to consider the difference between the canonical decomposition before and after the update. All of the slices corresponding to the higher-priority defaults are left unchanged by the update. Why? The new update default(s) is irrelevant context (in the sense of Observations 4.8) relative to those slices, because it is lower-priority relative to them. There is a **subtlety**, however. **The global prioritization even among the previous defaults may change: by transitivity "through" the new default(s).** According to the prioritization information in the update, one of the previous defaults (say, j) may be higher than a new default (say, k), and another (say, l) lower than that same new default (k), without there previously having been any strict priority between the previous two. That is, j may have been incomparable to l previously, yet be strictly higher after the update. Nevertheless, by exploiting, in effect, the monotonicity of prioritization (Theorem 2.58), we are able to show that the slices corresponding to the higher-priority defaults are affected monotonically by the update. The details are a bit tricky; see Appendix for the (unfortunately somewhat tedious) details.

□

Special Case: In sub-section 5.13.1, we show, as a corollary, that a default or module update is **globally monotonic** when the update is **lowest-priority**. That is, in this case, all of the previous conclusions are safe.

Corollary 5.56 (Higher-Priority for a Sequence of Updates)

In PDC and CLD: A sequence of default updates is partially monotonic with respect to all conclusions derived solely from the defaults previous to, and higher-priority than, all of those updates. More generally, a sequence of module updates is partially monotonic with respect to all conclusions derived solely from the modules (e.g., defaults) previous to, and higher-priority than, all of those updates.

Proof : Apply Theorem 5.55 to each update in the sequence. □

Observation 5.57 (Subtlety in Higher-Priority Conclusions)

There is a subtlety in the notion of a higher-priority conclusion, however. An "obvious", simplistic approach to defining the notion of a higher-priority conclusion is to:

Take all of the previous defaults that have higher priority than the new default, along with the prioritization among these, the global base, and the global fixtures. Then call any conclusion derivable from this axiom set a "higher-priority conclusion".

This **simplistic approach** does *not* work: it **is unsound**, in general. Why? The answer is that there may be *other* previous defaults that are *not* higher-priority than the update, but that *conflict*, in a manner unresolved by priority, with some of those that *are* previous higher-priority defaults. When prioritization is **non-layered**, this kind of situation can arise. Indeed, it can arise when there are only three defaults. The layered case, from which you might draw your intuitions, is simpler: such a situation cannot arise. In the layered case, every previous default that is *not* higher-priority than the update must have lower priority than all of those that *are* higher-priority than the update.

When prioritization is non-layered, there may be previous "spoiler" defaults that are incomparable-priority relative both to the new default and to some of the previous higher-priority defaults. Indeed, in the applications we have looked at, this happens commonly. And there may be conflicts, unresolved by priority, between these spoiler defaults and the higher-priority defaults. Thus there is a need to define derivability from the higher-priority defaults in such a way as to take into account this interaction context. That is, there is a **need for guarding** of these spoiler

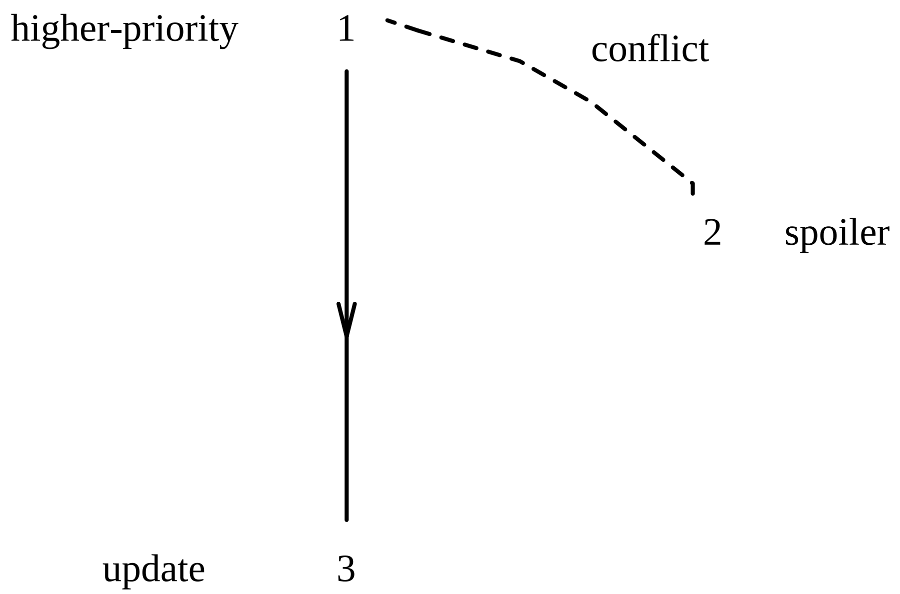


Figure 5.3: Subtlety in Notion of Higher-Priority: can arise even with just three defaults, when prioritization is non-layered.

defaults' default formulas when the higher-priority defaults are operating. By contrast, when the global prioritization is layered, any putative spoiler has its attempt at conflict squelched by priority.

More formally: Let R be the prioritization partial order after the update. Let u be the index of the new update default. The higher-priority defaults are those with indices in $R_D(u)$. Spoiler defaults are those defaults whose indices are in the set:

$\{j \in R_I(u) | \exists k.\ (k \in R_D(u)) \wedge (j \in R_I(k))\}$

Figure 5.3 illustrates this situation with just three defaults.

Our **solution** to this subtlety is that our notion of higher-priority conclusion is defined in terms of canonical derivability (Definitions 5.49 and 5.50).

A way to view our solution is that canonical derivability answers the question: Which of the conclusions sanctioned by the above simplistic approach were indeed actually sanctioned conclusions of the previous global theory? Interestingly, those that were, are exactly those that are guaranteed safe by our theorem.

Next, we give an example that illustrates our result about the safety of the higher-priority conclusions, which manifests some of the complexities that non-layered-ness brings. In particular, there is a default that plays the role of a "spoiler". We believe that the complexities that arise are representative of issues that will arise in large knowledge bases of expressively-rich defaults, e.g., information originating from common sense and natural language.

Example 5.58 (Traffic and Annoyances)

Lea, a retired woman, has various default and for-sure beliefs, represented in CLD. These beliefs provide a basis for her actions and planning. (Think of the CLD axioms below as someone else's partial information about her beliefs; then our grammatical tenses and scenario will make more sense. Note that we summarize as single premises some things that would be more accurately represented as implications or entailments. We also use a simplified representation of time.)

She has a default that parking tickets should be avoided: they are an annoyance.

$(d1) :> \ avoid\_parking\_ticket(day)$

She has a default that if she has to deal with a government bureaucracy on a given day, she should go first thing in the morning, to minimize the annoyance of the long lines that develop there. Getting there always involves driving.

$(d2) :> \ deal\_with\_bureau(day) \supset drive\_at\_time(day, 8amish)$

(By $8amish$, we mean roughly at 8am, i.e., plus or minus half an hour.) Lea also has a default that (usually) when going to an appointment at her medical (health maintenance organization) clinic, it is most convenient to drive there immediately beforehand. (It is a short driving distance from home, but not near any other places she ordinarily goes.)

$(d3) :> \ clinic\_appointment(day, time) \supset drive\_at\_time(day, time)$

Finally, Lea has a default that if the weather is bad and she has the flu, she should stay indoors in the morning. It is an annoyance to have to go outdoors then, especially since she is not a morning person in any event.

$(d4) :> \ weather(day, bad) \wedge health(day, flu) \ \supset \ stay\_indoors\_in\_am(day)$

In addition, Lea's for-sure beliefs include the following:

Avoiding parking tickets necessitates obeying parking regulations:

$\bullet\!> \ \forall day.\ avoid\_parking\_ticket(day) \supset obey\_parking\_regulations(day)$

Rush hours are: non-Sundays at 5pm or 6pm, and weekdays at 8am.

$$\begin{aligned}\bullet\!> \ \forall day.\ & (drive\_at\_time(day, 5pmish) \vee drive\_at\_time(6pmish)) \\ & \wedge \neg Sunday(day) \\ & \quad \supset \ drive\_in\_rush\_hour(day)\end{aligned}$$

$$\begin{aligned}\bullet\!> \ \forall day.\ & drive\_at\_time(day, 8amish) \\ & \wedge \neg Saturday(day) \wedge \neg Sunday(day) \\ & \quad \supset \ drive\_in\_rush\_hour(day)\end{aligned}$$

Going downtown to do major shopping at about 4pm and obeying the parking regulations there implies having to drive during rush hour, since the only parking is a 2-hour zone, and major shopping takes at least an hour.

$$\begin{aligned}\bullet\!> \ \forall day.\ & major\_shopping\_downtown(day, 4pmish) \\ & \wedge obey\_parking\_regulations(day) \\ & \quad \supset \ (drive\_at\_time(day, 5pmish) \vee drive\_at\_time(day, 6pmish))\end{aligned}$$

Driving early in the morning makes her feel tired then (again, she is not a morning person).

$\bullet\!> \ \forall day.\ drive\_at\_time(day, 8amish) \supset feel\_tired(day, 8amish)$

Various facts about weather, health, clinic appointments, obligations to deal with the bureaucracy, major shopping trips to downtown, and days of the week:

(Notation: As a shorthand for conjunctions of for-sure assertions of positive or negative literals, we list the satisfying objects, or, more generally, tuples. Often, in this context, we use "..." to indicate that there are additional satisfying tuples not shown explicitly; for simplicity's sake, we assume these objects are distinct from all other explicitly-shown objects.)

$\bullet\!> \; \neg Sunday : 8/15/86, 8/16/86, 8/17/86, \ldots,$
$\qquad 11/10/86, 11/11/86, 11/12/86, 11/13/86, \ldots$

$\bullet\!> \; \neg Saturday : 8/15/86, 8/16/86, 8/17/86, \ldots,$
$\qquad 11/10/86, 11/11/86, 11/12/86, 11/13/86, \ldots$

$\bullet\!> \; +weather : \langle 11/10/86, good\rangle, \langle 11/11/86, bad\rangle,$
$\qquad \langle 11/12/86, cloudy\rangle, \langle 11/13/86, good\rangle, \ldots$

$\bullet\!> \; \forall day.\, \neg(weather(day, bad) \wedge weather(day, good))$

$\bullet\!> \; +health : \langle 11/10/86, fine\rangle, \langle 11/11/86, flu\rangle,$
$\qquad \langle 11/12/86, flu\rangle, \langle 11/13/86, fine\rangle, \ldots$

$\bullet\!> \; \forall day.\, \neg(health(day, flu) \wedge health(day, fine))$

$\bullet\!> \; +clinic\_appointment : \langle 8/15/86, 5pmish\rangle, \langle 11/12/86, 2pmish\rangle, \ldots$

$\bullet\!> \; +deal\_with\_bureau : 8/16/86, \ldots$

$\bullet\!> \; +major\_shopping\_downtown : \langle 11/13/86, 4pmish\rangle, \ldots$

Finally: one particular day she came down with the flu while visiting a friend overnight on a weeknight, after parking in a commercial zone not legal for her car after 8am on weekdays. Unfortunately, it was also bad weather that next morning. Thus she found herself in the position of having either to go out in the morning with the flu or to disobey the parking regulations.

$\bullet\!> \; \neg(stay\_indoors\_in\_am(11/11/86) \wedge obey\_parking\_regulations(11/11/86))$

Degree of likelihood-weighted annoyance, i.e., decision-theoretic utility, is the basis for precedence between Lea's defaults. The first four defaults did not have relative precedence established between them. Parking tickets, going outdoors early when sick, and waiting at the bureaucracy all involve roughly equal amounts of annoyance. And she is able to choose her own times for clinic appointments (it provides walk-in service), hence she does not anticipate potential conflict between the clinic default and the others.

Now, suppose one updates the above axiom set with the new default that Lea does not drive in rush hour: it is an annoyance to her.

$(d5) :> \; \neg drive\_in\_rush\_hour(day)$

Lea can anticipate that not driving in rush hour will conflict with some of her other defaults.

For example, she knows that major shopping downtown in the afternoons can lead to driving in rush hour unless she disobeys the parking regulations, which implies asking for a parking ticket (her town has viciously efficient enforcement). But getting a parking ticket is significantly more of an annoyance than driving in rush hour. Thus she gives the no-rush-hour default lower prioritization than the default to avoid parking tickets.

$\mathcal{PREFER}(d1, d5)$

And she knows that going early to deal with the government bureaucracy involves driving in rush hour. But getting stuck in the lines there is significantly more of an annoyance than driving in rush

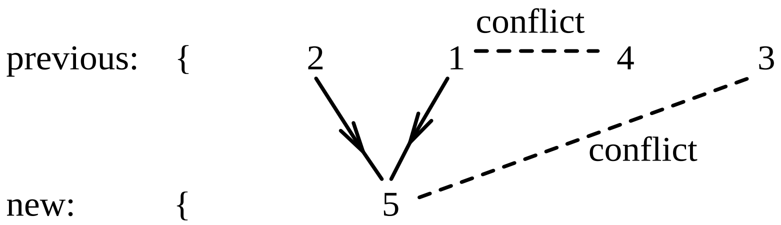


Figure 5.4: Traffic and Annoyances: after the update. $d1$ and $d2$ have higher priority than $d5$, the new default. There is conflict, unresolved by priority: between the pair $d1$ and $d4$, and between the pair $d3$ and $d5$. Note that the global prioritization is non-layered. $d4$ plays the role of a "spoiler" default.

hour. Thus she gives the no-rush-hour default lower prioritization than the default to go early to deal with the bureaucracy.

$\mathcal{PREFER}(d2, d5)$

By contrast, she does not give any strict precedence between the new no-rush-hour default and the other two previous defaults. She does not anticipate potential conflict between the new default and the stay-indoors default. And if she is sick in bad weather, driving is unpleasant whether it is in the rush hour or not. Insofar as she can foresee conflict between the new default with the clinic default, the degree of annoyance involved is roughly equal. Hence, she has no strong, i.e., definitive, preference as to how to resolve any conflicts with those two previous defaults.

The default update thus consists of exactly the last three axioms listed above.

Then what does our result the safety of higher-priority conclusions (Theorem 5.55) say, for this example? Figure 5.4 illustrates the global prioritization, and the unresolved conflicts, after the update.

The set of higher-priority previous defaults is:

$\{d1, d2\}$

i.e., the parking-ticket and bureaucracy rules. Thus our result guarantees that all of the conclusions derived solely from $\langle d1, d2\rangle$ (i.e., the conjunction of their corresponding slices) are preserved by the update. In particular, it **guarantees** that:

The previous conclusions

$obey\_parking\_regulations(8/15/86)$

$obey\_parking\_regulations(8/16/86)$

$obey\_parking\_regulations(8/17/86)$

$obey\_parking\_regulations(11/10/86)$

$obey\_parking\_regulations(11/12/86)$

$obey\_parking\_regulations(11/13/86)$

that were derived solely from $d1$ (parking-ticket) are safe. These do not involve any conflict with the new (rush-hour) default $d5$.

Moreover, the previous conclusions

$drive\_in\_rush\_hour(8/16/86)$

$feel\_tired(8/16/86, 8am)$

that were derived solely from $d2$ (bureaucracy) are safe. These do indeed involve conflict with the new (rush-hour) default $d5$, but the conflict is resolved by priority.

Likewise, the previous conclusion

$drive\_in\_rush\_hour(11/13/86)$

that was derived solely from $d1$ (parking-ticket, via the major shopping trip), and does involve conflict with the new default, is safe. The conflict is resolved by priority.

So much for the guarantees of safeties. Next, we remark on non-safeties in this example:

By contrast, the previous conclusion

$drive\_in\_rush\_hour(8/15/86)$

is retracted. It was derived from $d3$ (clinic), but involves conflict with the new default. The conflict is not resolved by priority: $d3$ and the new default are incomparable. The default update is **not globally monotonic**.

The subtlety in defining the higher-priority conclusions, that we remarked upon in Observation 5.57, is illustrated in this example by the conflict, unresolved by priority, between the previous defaults $d1$ (parking-ticket) and $d4$ (stay-indoors). Relative to the update: $d1$ is higher-priority, and $d4$ plays the role of a **"spoiler"** (in the sense of Observation 5.57). $d4$ is not higher-priority; it has priority incomparable both with the update and with a previous higher-priority. $d1$ and $d4$ conflict about the day 11/11/86. On that day, recall, Lea was sick in the morning and had to choose between going outdoors or disobeying the parking regulations. Thus

$obey\_parking\_regulations(11/11/86)$

is not a higher-priority conclusion. Actually, it was not a previous conclusion at all. Yet, in the simplistic approach, it is indeed a "higher-priority" conclusion. By contrast: in the slice where $d1$ is maximized, $d4$'s default formula is guarded, and this prevents concluding it.

Finally, we observe that, in this example, there are safeties present that are *not* captured by Theorem 5.55. For example, the previous conclusion

$drive\_at\_time(11/12/86, 2pmish)$

is preserved under the update. It is derived from $d3$ (clinic). Since $d3$ is not higher-priority than the new default, there may, in general, be conflict between it and the update; but this particular conclusion did not happen to involve any.

In chapters 6 and 7, we will discuss additional kinds of guarantees of safeties. These kinds go beyond higher prioritization, and apply to many cases of such incomparable priority and non-conflict.

Next, we give another more-than-small example of safeties of updating based on higher prioritization. Two features are especially interesting. Firstly, there is a topical group of higher-priority defaults. Secondly, the prioritization structure, though non-layered, nevertheless prevents the possibility of "spoiler" defaults (in the sense of Observation 5.57).

Example 5.59 (Safety of Higher-Priority, Extended Meetings)

We extend the Meetings example (Figure 2.1 and section 2.8), in CLD form. We modify the example in several ways, including: we add some more emergency-module defaults; we add a bunch of base axioms, e.g., about particular emergencies; we change the names and arguments of the predicates somewhat; and we make the person type explicit.

Again, as a shorthand for conjunctions of for-sure assertions of positive or negative literals, we list the satisfying objects, or, more generally, tuples. In this context, we use "..." to indicate that there are additional satisfying tuples not shown explicitly; for simplicity's sake, we assume these objects are distinct from all other explicitly-shown objects.

We begin with the original Meetings example defaults.

The emergencies module consists of:

$(d1) :> \ fire(place, day) \wedge person(x) \ \supset \ leave(x, place, day)$

The meetings module consists of:

$(d3) :> \ in\_group(p, 4321) \wedge Tuesday(d) \ \supset \ group\_meeting(p, BldgA, d)$

$(d2) :> \ in\_group(p, 4321) \wedge vacation(Boss(4321), d)$
$\qquad \supset \ \neg group\_meeting(p, BldgA, d)$

$\mathcal{PREFER}(d2, d3)$

$\mathcal{PREFER}(d1, d2)$

The attendance module consists of:

$(d6) :> \ weekday(d) \wedge reg\_employ(person, place)$
$\qquad \supset \ attend\_work(person, place, d)$

$(d4) :> \ flu(person, day) \ \supset \ \neg attend\_work(person, place, day)$

$(d5) :> \ holiday(day) \ \supset \ \neg attend\_work(person, place, day)$

$\mathcal{PREFER}(d4, d6)$

$\mathcal{PREFER}(d5, d6)$

$\mathcal{PREFER}(d1, d4)$

$\mathcal{PREFER}(d1, d5)$

We add some base axioms to flesh things out:

$\bullet\!> \ \forall x, place, day.\ leave(x, place, day) \supset \neg attend\_work(x, place, day)$

$\bullet\!> \ \forall p, d.\ group\_meeting(p, d) \wedge in\_group(p, 4321)$
$\qquad \supset \ attend\_work(p, BldgA, d)$

$\bullet\!> \ +person: \ Fred, Pearl, Tim, Tom, Chang, Sue,$

$\bullet\!> \ +person: \ Eileen, Andy, Ed, Peg, Maggie, \ldots$

$\bullet\!> \ +weekday: \ 3/12/91, \ldots$

$\bullet\!> \ +reg\_employ: \ \langle Ed, BldgA \rangle, \ldots$

$\bullet\!> \ +flu: \ \langle Ed, Today \rangle, \ldots$

$\bullet\!> \ +holiday: \ 7/4/90, \ldots$

$\bullet\!> \ \neg holiday: \ 3/12/91, Today, 7/4/90, 11/12/91, 4/1/90, \ldots$

$\bullet\!> \ +Tuesday: \ Today, 11/12/91, 4/1/90, \ldots$

- •> $\neg Tuesday:\;\; 3/12/91, \ldots$
- •> $\forall d.\, Tuesday(d) \supset weekday(d)$
- •> $+in\_group(p, 4321):\;\; Ed, Peg, Maggie, \ldots$
- •> $\forall person.\, in\_group(person, 4321) \supset reg\_employ(person, BldgA)$
- •> $+vacation(Boss(4321), d):\;\; 11/12/91, \ldots$
- •> $\neg vacation(Boss(4321), d):\;\; Today, \ldots$
- •> $\neg attend\_work:\;\; \langle Peg, 6/4/90\rangle, \langle Maggie, 7/5/91\rangle, \ldots$
- •> $\neg group\_meeting:\;\; \langle Ed, 1/3/90\rangle, \langle Peg, 1/3/90\rangle, \langle Maggie, 1/3/90\rangle, \ldots$

We add some more emergencies defaults, for earthquakes, tornadoes, and military attacks:

$(d7) :>\; earthquake(place, day) \wedge person(x) \;\supset\; leave(x, place, day)$

$(d8) :>\; tornado(place, day) \wedge person(x) \;\supset\; leave(x, place, day)$

$(d9) :>\; military\_attack(place, day) \wedge person(x) \;\supset\; leave(x, place, day)$

$\mathcal{PREFER}(d7, d2)$

$\mathcal{PREFER}(d7, d4)$

$\mathcal{PREFER}(d7, d5)$

$\mathcal{PREFER}(d8, d2)$

$\mathcal{PREFER}(d8, d4)$

$\mathcal{PREFER}(d8, d5)$

$\mathcal{PREFER}(d9, d2)$

$\mathcal{PREFER}(d9, d4)$

$\mathcal{PREFER}(d9, d5)$

We add some base axioms about particular emergencies:

- •> $fire(Baltimore, 2/4/03)$
- •> $fire(Watts, 8/2/67)$
- •> $fire(BldgA, 4/1/90)$
- •> $earthquake(SF, 4/8/06)$
- •> $earthquake(MexicoCity, 5/3/87)$
- •> $tornado(Omaha, 7/16/75)$
- •> $tornado(Milford, 8/15/89)$
- •> $military\_attack(Rome, 6/17/43)$
- •> $military\_attack(Baghdad, 1/21/90)$
- •> $\forall x, place, day.\, hurt(x, day) \supset \neg leave(x, place, day)$
- •> $hurt(Chang, 4/8/06) \wedge hurt(Sue, 4/8/06)$
- •> $hurt(Fred, 8/15/89) \equiv hurt(Pearl, 8/15/89)$
- •> $hurt(Tim, 8/2/67) \vee hurt(Tom, 8/2/67)$

And we have some other miscellaneous information not particularly oriented towards the current set of defaults. E.g., about various people being at professional conferences on various days:

- •> $+conferee : \langle Maggie, 3/12/91\rangle, \ldots$

Now consider this extended Meetings example as a previous global axiom set. It has quite a few

NM conclusions!

Suppose one receives the default update that another overriding exception to the weekday rule is that: if one is attending a professional conference, then one is not at one's usual workplace. This new default goes into the attendance module. It has lower prioritization than the emergencies module, but greater prioritization than the weekday rule. In CLD, the update consists of the axioms:

$$(d10) :> \ conferee(x, day) \wedge person(x) \wedge reg\_employ(x, place)$$
$$\supset \ \neg work(x, place, day)$$

$\mathcal{PREFER}(d1, d10)$
$\mathcal{PREFER}(d7, d10)$
$\mathcal{PREFER}(d8, d10)$
$\mathcal{PREFER}(d9, d10)$
$\mathcal{PREFER}(d10, d6)$

The update causes some retractions, e.g., the previous conclusion

$attend\_work(Maggie, 3/12/91)$

Then the set of higher-priority defaults is exactly the emergencies module. Theorem 5.55 thus guarantees the safety of all the previous conclusions derived solely from the emergencies-module's defaults, i.e., based only on those defaults plus the for-sure axioms. For example, the previous NM conclusions:

$leave(Eileen, SF, 4/8/06)$
$leave(Andy, Omaha, 7/16/75)$
$leave(Ed, BldgA, 4/1/90)$
$\neg attend\_work(Maggie, BldgA, 4/1/90)$
$\neg group\_meeting(Ed, BldgA, 4/1/90)$

are all guaranteed to be safe.

Observe that in this example, even though the global prioritization is non-layered, there are no "spoilers" cf. Observation 5.57. The higher-priority defaults form a clean module slice: this motivates our next results.

5.13 Total Prioritizations

In this section, we show that in many special cases, decompositions along prioritization are guaranteed to produce slices or phases that *are* clearly simpler than the global.

We show (sub-section 5.13.1) that one such special case is when there is total (Definition 2.53) prioritization between modules. Then there is always a *clean* slice or phase: with fewer defaults, and with base the same as the global's.

A yet more special case is when all defaults are propositional, and there is total prioritization among them. We show (sub-section 5.13.2) that in this case a particularly simple form of serial decomposition obtains, in which there is no need to protect (fix or guard). Furthermore, each phase is just equivalent to a simple consistency check for that phase's default. This yields a **new forward-inference algorithm**. And it is suggestive for more algorithms and for comparison to other NM formalisms defined in terms of serial consistency. This result is less obvious than it first appears. We show that there is a subtlety: no global fixtures may be allowed.

In addition, we show (sub-section 5.13.3) that one can view base axioms as a special case of prioritized default axioms: as defaults at highest-priority. Updating with base axioms is thus, essentially, a special case of updating with prioritized default axioms. This supports our effort to formalize (prioritized) defaults as axioms in circumscription. And it reinforces our interest in the problem of updating with new defaults.

5.13.1 Totality Over Modules

Suppose the global defaults (e.g., minimized predicates) can be split into two groups, where all of the defaults in the first group have higher priority than all of the defaults in the second.
Terminology: We call this splitting: **partition** of the global prioritization.
Then one can form a *clean* slice or phase that corresponds to the higher-priority group, i.e., the higher-priority module (Definition 2.62). This follows from the 2-ary strict-priority, module case of canonical decomposition. The next two corollaries give precise details.

Corollary 5.60 (Decomposition by 2-ary Priority Among Modules)
Suppose $H = (MOD; RE)$, where $MOD = \langle MOD1, MOD2 \rangle$, and RE contains exactly the pair $(1,2)$.
Then, for any base B:

$$MC(B; H; Z) \equiv MC(B; MOD1; Z) \wedge MC(B; MOD2; fix\ MOD1; Z)$$

and

$$MC(B; H; Z) \equiv MC(MC(B; MOD1; Z); MOD2; fix\ MOD1; Z)$$

More generally, for any base B, global fixture F, and global guard G:

$$\begin{aligned} &MC(B; H; fix\ F; guard\ G; Z) \equiv \\ &\quad MC(B; MOD1; fix\ F; guard\ G; Z) \\ &\quad\quad \wedge\ MC(B; MOD2; fix\ MOD1, F; guard\ G; Z) \end{aligned}$$

and

$$\begin{aligned} &MC(B; H; fix\ F; guard\ G; Z) \equiv \\ &\quad MC(MC(B; MOD1; fix\ F; guard\ G; Z); MOD2; fix\ MOD1, F; guard\ G; Z) \end{aligned}$$

(Recall the MC notation from Definition 5.27 of module circumscription.)

Proof : This is the special 2-ary, strict-priority case of: the module case of canonical decomposition along prioritization (Corollary 5.28 and the module case of Theorem 5.32). □

But observe that fixing a prioritized default pre-order is just equivalent to fixing each of its default pre-orders, in parallel. Thus fixing $MOD1$ is equivalent to fixing $D1$.

Corollary 5.61 (Decomposition by 2-ary Priority Among Groups)

In PDC or CLD:
Suppose the global prioritized default pre-order $H = (D; R)$, when viewed as a super-module, is composed of two modules $MOD1 = (D1; R1)$ and $MOD1 = (D2; R2)$, where $MOD2$ has higher priority than $MOD2$, as in Corollary 5.60. Let B be the global base, F be the global fixture Then

$$PDC(B; D; R; fix\ F; Z) \ \equiv$$
$$PDC(B; D1; R1; fix\ F; Z) \ \wedge \ PDC(B; D2; R2; fix\ D1, F; Z)$$

and

$$PDC(B; D; R; fix\ F; Z) \ \equiv$$
$$PDC(\ PDC(B; D1; R1; fix\ F; Z) \ ; D2; R2; fix\ D1, F; Z)$$

More generally, global guards may be present, as in Corollary 5.60; they then appear in each of the slices or phases as well.

Proof : Immediate from Corollary 5.60, plus a lemma that fixing a prioritized pre-order is equivalent to fixing its starting pre-orders in parallel. (This lemma is discussed in the proof of Corollary 5.28.) □

Observation 5.62 (First Slice or Phase Is Clean; CLD View)
Total prioritization among modules is a well-behaved special case. In Corollary 5.61, the global can be conjunctively decomposed into two slices: one where the higher-priority group of defaults is maximized; and another where the lower-priority group of defaults is maximized. Each slice PDC, as well as the global PDC, corresponds to a CLD axiom set. The higher-priority slice axiom set $\mathcal{SA}_1$ consists of the global base axioms plus the global fixture axioms plus the higher-priority defaults' default axioms plus their associated internal prioritization's prioritization axioms. Compared to the global axiom set $\mathcal{A}$, $\mathcal{SA}_1$ is just missing some default and prioritization axioms. (The axioms associated with the lower-priority defaults are irrelevant context, in the sense of Observation 4.8, as far as the higher-priority defaults are concerned.) That is, the first constituent axiom set is a proper subset of the global axiom set:

$$\mathcal{SA}_1 \subset \mathcal{A}$$

Thus the first slice is **clean** (Definition 4.7), with all the potential advantages we observed in subsection 4.5.3. The first slice is **clearly simpler than the global**, as an axiom set and as a theory, according to a clear and, to our mind, indisputably valuable *criterion*: *fewer defaults and fewer axioms* (fewer in the sense of subset not just count). Unlike the general-case situation that we discussed in sections 5.6 and 5.11, **no protection conditions are needed** at all: no introduced fixtures, and no introduced base axioms. Therefore:

1. **Locality:** (Non-trivial) NM inference can be performed *locally* in a sound fashion, with whatever procedures are available for CLD, while working just with the smaller axiom set $\mathcal{SA}_1$.

2. **Selectiveness:** And even if that inference is performed exhaustively on $\mathcal{SA}_1$, it will generate only a *part* of the overall global theory.

We discuss inference procedures for CLD in chapter 7.

Likewise, the global can be serially decomposed into two phases, in descending sequence of priority: a first where the higher-priority group of defaults is maximized; and a second where the lower-priority group of defaults is maximized. The **first, higher-priority phase is clean** (Definition 4.16), and thus **clearly simpler than the global**. This first, higher-priority phase's axiom set is the same as the higher-priority slice's axiom set. Thus the higher-priority slice's theory is equivalent to the first, higher-priority phase's theory (i.e., the first tier).

By contrast, the second, lower-priority slice or phase is *not* clean. $\mathcal{SA}_2$ is a *mutation* (cf. discussion after Definition 4.1). Compared to the global axiom set, it is missing some default and prioritization axioms, but it *also* contains some new *fixture* axioms. However, even in this slice or phase, the protection conditions are simpler than in the general case of canonical decomposition, since there are no external incomparable-priority defaults requiring guarding. Thus the base in the second slice is just the global base.

Previous Results:
For Corollary 5.61, the relevant previous results are by Lifschitz: Theorems 5.1 and 5.2. He showed, for the predicate case, that one could decompose by *layers*, both conjunctively and serially. Here, we showed that one can **decompose by modules** which may have internal prioritization. Moreover, unlike his results, in ours, both the global prioritization and the internal prioritization within each module (i.e, group of defaults or predicates) may be **non-layered**. By contrast, within a layer, the internal prioritization is parallel, i.e., empty. Lifschitz did not use the concept of composition of prioritization (it is new with us). If he had, the internal prioritization would have been restricted to be layered.

Example 5.63 (Emergencies Slice Is Clean)
Consider the extended Meetings example (Example 5.59). Then the emergencies module is a highest-priority group of defaults, and corresponds to a clean slice. For the basic Meetings example, we can see this cleanliness in Example 5.19. There the slice in which the fire default is maximized, i.e., in which its abnormality predicate $ab1$ is minimized, is clean. Note that in these Meetings examples, the global prioritization is non-layered.

Example 5.64 (Mikey's Sources of Beliefs)
Mikey is a child in fourth grade. Mikey gets all his information about the wide world either from his family or from what he learns in school. Mikey has a mother, a father, a sister, a brother, a set of school teachers, a set of school friends, and a set of school enemies. Each of these acquaintances tells him many things that he adopts as premise defaults. For example, his father tells him that adventurousness is good, but, as an exception, that extreme adventurousness (foolhardiness) is bad. Thus Mikey's defaults arising from any given source have precedence among them, in general. However, he trusts these various sources differently. He believes his parents more than anyone else. He believes his teachers more than his school friends. He believes his school enemies less than

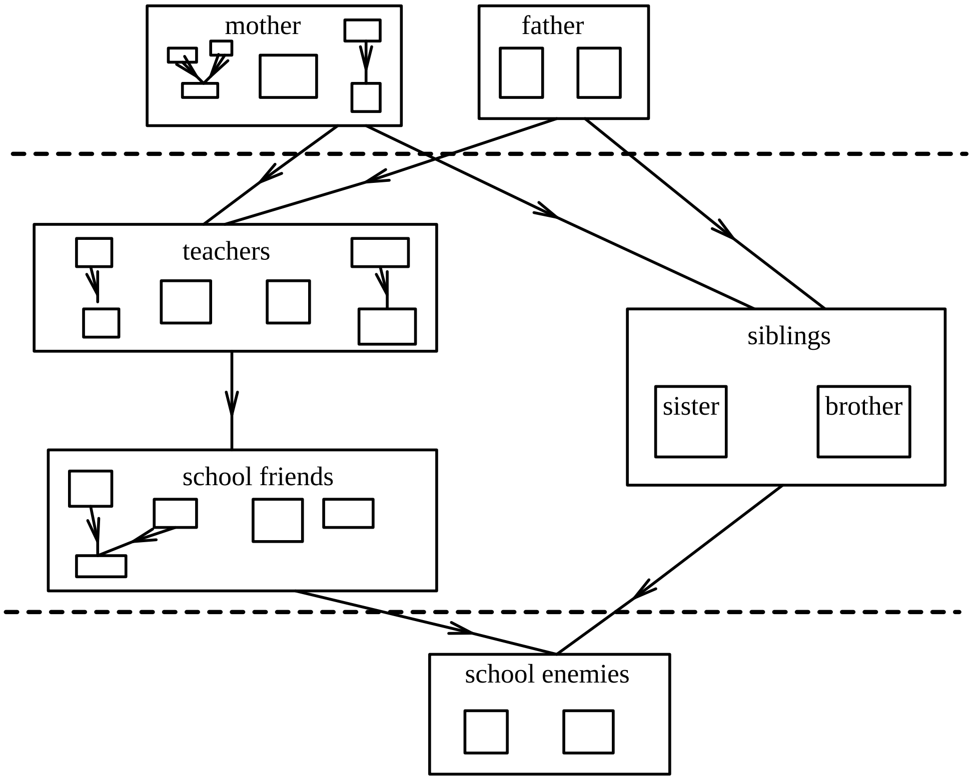


Figure 5.5: Mikey's Sources of Beliefs: Mikey's defaults can be represented in terms of modules associated with sources of information. Each box indicates a non-primitive module. Each dashed line indicates a possible choice of partition by prioritization, i.e., choice of division to create a pair of super-modules: one higher priority than the other.

anyone else. Some teachers he believes more than others; some school friends he believes more than others. We can represent this precedence structure among his default beliefs in terms of modules. Figure 5.5 illustrates.

In addition, Mikey has some for-sure premise beliefs.

There are thus two different possible ways to partition the global set of defaults by prioritization, in this example. By partition here, we mean to create a pair of super-modules, one higher priority than the other. One choice for the higher-priority super-module is to include just the mother and father modules: i.e., a parents super-module. Another choice is to include all but the school enemies module: i.e., a "friendlies" super-module. Each choice generates a clean higher-priority slice cf. Corollary 5.61.

The ability to choose among several possible available prioritization partitions in Example 5.64 illustrates a broader point. In general, for a given CLD axiom set, there may be many different ways to partition the prioritization.

Observation 5.65 (Hierarchical Choice in Clean Slicing)

Any n-ary priority chain of modules can be cut into two sub-chains at *any* point in the chain. (By chain here, we mean that the prioritization between the modules is a total order.) But these sub-chains are themselves modules. Thus, when a set of defaults has this prioritization structure, one can *hierarchically choose the desired grain size of clean slicing*: a larger or a smaller clean higher-priority slice cf. Corollary 5.61. This allows control over the selectiveness of exhaustively-applied inference, for example.

Partitionability of prioritization also immediately yields a nice result about safety of updating.

Corollary 5.66 (Monotonicity of Lowest-Priority Default Updates)
In CLD, suppose that an update consists of one or more default axioms and some prioritization axioms, such that the new defaults all have lower priority than all of the previous defaults, according to the prioritization after the update. Then *all* of the previous conclusions are safe under the update. I.e., the update is globally monotonic.

In other words, suppose a module update is such that the new module is lower priority than all of the previous defaults. Then all of the previous conclusions are safe under the update.

Proof : Immediate from Theorem 5.55: the set of higher-priority previous defaults is the entire set of previous defaults. □

Examples of Lowest-Priority Updates:
In the Mikey example (Example 5.64), let the previous axiom set consist of some base axioms plus all the modules except for the school-enemies module. Then updating with the school-enemies module is globally monotonic. In the Quaker-Republican example (Example 1.5): consider beginning with the Republican default plus some for-sure information about Republicans and Quakers and Pacifists. Let an update consist of the Quaker default plus the prioritization information that this new default is lower-priority than the previous Republican default. Then that default update is globally monotonic.

Corollary 5.67 (Updating in a Single Column)
A special case of Corollary 5.66 is:
In a PDC, suppose the global prioritization has the form of a single column, e.g., representing a default inheritance chain. Then updating with a new default whose prioritization is lowest-priority, e.g., least specific, is globally monotonic.

Example 5.68 (Updating With Least Specific Default, Bad Weather)
Consider a global CLD axiom set about J's commuting times: it contains some for-sure axioms plus two defaults. The first says that J's commute at night in bad weather takes between 60 and 80 minutes. The second says that in heavy snow at night it takes at least 90 minutes. Heavy snow is a kind of bad weather. The second default is more specific, and has greater priority.

$$(d1) :> \ commute(J, trip) \wedge bad\_weather(trip) \wedge night(trip) \\ \supset \ length(trip, 60\_to\_80)$$

$$(d2) :> \ commute(J, trip) \wedge heavy\_snow(trip) \wedge night(trip)$$

$\supset\ length(trip, more\_than\_90)$

$\mathcal{PREFER}(d2, d1)$

Suppose we update this axiom set with a new default. This third default says that J's commute is (ordinarily) 50 minutes. This default is less specific than the previous two, and has lower priority than both of them.

$(d3) :>\ commute(J, trip)\ \supset\ length(trip, 50)$

$\mathcal{PREFER}(d1, d3)$

Then this default update (consisting of these last two axioms) is globally monotonic.

5.13.2 Propositional Defaults; Collapse

A yet more special case of totality over modules is when all defaults are propositional, and there is total prioritization among them. Next, we show that in this case a particularly simple form of serial decomposition obtains, in which there is no need to protect (fix or guard) at all. **Every phase is clean.** Furthermore, each phase is just equivalent to a simple consistency check for that phase's default. This yields a **new forward-inference algorithm**. And it is suggestive for more algorithms and for comparison to other NM formalisms defined in terms of serial consistency.

Intuitively, totality of the prioritization over propositional defaults completely rules out the potential for conflict between defaults. Thus **locality** (in the sense of sections 4.5 and 4.6) is as great as possible.

This result is less obvious than it first appears, however. We show that there is a subtlety: no global fixtures may be allowed.

Theorem 5.69 (Clean Serial, Propositional Defaults)
In a propositional prioritized default circumscription $PDC(B; D; R; Z)$ without fixtures:
(By propositional, we mean that the base logical language is propositional.)
Suppose that the prioritization is a total order (cf. Definition 2.53).
Then the global PDC is serially decomposable into n *clean* phases, one per default, in each of which exactly that default is maximized. The sequencing is descending by priority. I.e., the first phase is the highest-priority default, and the last is the lowest-priority default.

$$PDC(B; D; R; Z)\ \equiv\ \underset{\substack{i \in N \\ via \\ R}}{///}\ PDC(\underset{B}{\bullet}\ ; Di; \emptyset; Z)$$

where N is the index set of D. (Notationally, in the right-hand side PDC expression, we could have omitted the prioritization, which is empty, since there is only one default there.)

Moreover, each phase is just equivalent to a simple consistency check for that phase's default. That is, for each $i \in N$:

$$PDC(BBi; Di; \emptyset; Z)\ \equiv\ BBi[Z]\ \wedge\ \{(\exists Z'.\, BBi[Z'] \wedge Di[Z'])\ \supset\ D[Z]\}$$

where BBi is the base for the i^{th} phase. (For $i > 1$, BBi is the tier resulting from the $(i-1)^{th}$ phase.) Thus the global PDC can be computed with the following

Decidable Algorithm for Exhaustive Forward Inference:

0) Number the defaults' priority from 1 as highest to n as lowest. 1) Begin with $\mathcal{TIER}$ assigned to B. 2) For i from 1 to n: Test the consistency of Di against $\mathcal{TIER}$ using some satisfiability algorithm. (Note that propositional satisfiability is NP-hard.) If Di is consistent, revise $\mathcal{TIER}$ to include (i.e., conjoin) Di. (Then go on to the next i.) 3) The resulting $\mathcal{TIER}$ is the global circumscription. That is, it axiomatizes the circumscription, in the propositional base logical language $\mathcal{L}$: no second-order quantifiers involved. By saying that the algorithm performs "exhaustive" forward inference, we mean exhaustive in the non-monotonic aspect of reasoning. Only monotonic inference is required to use the result of the algorithm to arrive at any NM conclusion in the NM theory.

Generalization to Closed Defaults in First-Order Base Languages: This result generalizes to first-order (and, indeed, higher-order) base languages $\mathcal{L}$: as long as the defaults are all closed. That is, the default formulas must all be closed formulas.

The base sentence B is not required to be propositional; e.g., it may mention first-order (or even higher-order) quantifiers.

However, there is a further **subtlety: even functions may not be fixed**; this is an exception to our **assumption** (page 36) that all functions are fixed. This requirement is not terribly onerous. Including a complete theory of equality, e.g., domain closure plus the uniqueness of all names, in the base sentence suffices to effectively, though indirectly, fix the function symbols in a useful way, for example: see Theorem 2.24. Domain closure plus the uniqueness of all names suffices for the above result, for example, in place of no fixing of function symbols. However, in this case, note that the expressive power of the base language is effectively constrained to be no more than propositional. (By "effectively", we mean insofar as the circumscription is concerned.)

Of course for (full) first- and higher- order logic, consistency is not decidable. Thus the above algorithm becomes a procedure. The result of the procedure (if there is one) is an axiomatization of the circumscription, in the same logical language used to express the base sentence: e.g., an axiomatization in first-order if the base and default formulas were expressed in first-order.

Collapse to Propositional or First-Order: In short, in this case, circumscription **collapses** to the same expressive class as the base language $\mathcal{L}$ used to express the base and default formulas.. The circumscription is propositional if the base language is; it is first-order if the base language is.

Proof Overview: Non-trivial. Uses Corollary 5.61. Inductive, in the ascending direction of priority. Uses also three lemmas about how the propositional case behaves nicely. See Appendix. □

Discussion: No Fixtures:

Intuitively, one way to view the requirement that no fixtures are permitted is that when fixture is represented as part of the maximized pre-order, it is in parallel with the rest of the maximized pre-order that represents the defaults. But this violates totality of the prioritization. Thus implicit fixture arising from extreme conflict (Definition 3.53) is also prohibited by the requirement of totality; remember that in Lemma 3.54, the polar defaults must be in parallel with each other.

Observation 5.70 (Non-Superfluity of the Fixture Prohibition)

The following counterexamples to the equivalence of even a single phase to a consistency test shows the non-superfluity of the prohibition on fixture in Theorem 5.69. In each counterexample, there is a single default, and a single fixture.

In the propositional-language case: let B be $True$, i.e., the empty conjunction; let D be the proposition p; and let the fixture F also be p. Then the default p is consistent with the base. But the circumscription is equivalent to the base $True$, since the fixture blocks the default.

In the first-order-language case: let

$$B \stackrel{\text{def}}{\equiv} [q \equiv (a = b)]$$

where q is a proposition and a and b are distinct 0-ary function symbols. Let D be q. Then the default q is consistent with the base. But the circumscription is equivalent to the base, since the fixing of the function symbols blocks the default.

Observation 5.71 (Discussion: Why Serial Is Clean)

There is another perspective on why the serial decomposition in Theorem 5.69 is clean. In each phase, the current base (i.e., the immediately previous tier) effectively fixes all of the previous (higher-priority) defaults' default formulas. In terms of Corollary 5.61, this obviates the need for the introduced fixture of the higher-priority defaults ($D1$ in that Corollary) in the lower-priority-module's phase (or slice). By "effectively fixes" here, we mean that the previous defaults' default formulas are fixed relative (Definition 3.47) to the current base. Why are they are effectively fixed? In the i^{th} phase, either the base implies the negation of the i^{th} default formula Di, or else that phase concludes Di. Thus, in the following phases, the base (for those phases) either implies $\neg Di$ or it implies Di. This effectively fixes Di in those later phases. But this is true for each i.

Observation 5.72 (Discussion: Algorithms)

Negation by failure: In the predicate case of Theorem 5.69, checking for consistency is exactly: negation as failure. I.e., the negation of a minimized proposition abi is concluded when one fails to find a proof of abi from $\mathcal{TIER}$ when using a (sound and) complete monotonic inference method for the base language.

Selectiveness: In Theorem 5.69, if one stops the series / algorithm after the k^{th}-highest priority default, then one obtains a sub-theory ($\mathcal{TIER}_k$) of the global theory. This sub-theory is a clean higher-priority slice / phase, corresponding to the k highest priority defaults, cf. Corollary 5.61.

Comparison to Theorem 5.29:

Like Theorem 5.29 (an open default is equivalent to the parallel prioritization of its instances), Theorem 5.69 is at instance-grain. However, here the prioritization is total rather than empty.

Observation 5.73 (Prioritization Versus Inferential Precedence)
Theorem 5.69 (even more than the serial case in Corollary 5.61) is suggestive in another regard. Each default is simply invoked if it is consistent; but the higher-priority defaults "get their say", i.e., are maximized, before the lower-priority defaults do. This corresponds to the intuition that higher priority means *inferential precedence* (**terminology**) in the sense of sequencing: non-monotonic conclusions from the higher-priority defaults are drawn *before* going on to deal with the lower-priority defaults.

It is because the prioritization is total not just over open defaults, but also over default instances that the picture is simple. By contrast, the descending-sequence result available in Corollary 5.61 requires the higher-priority defaults' default formulas to be fixed[7] in the later, lower-priority phases.

The correspondence between prioritization and inferential precedence is more complicated still in the case of more general non-layered prioritization partial orders. The protection conditions in Theorem 5.32 (descending serial, general) manifest this complexity. Even in the propositional case: if there is total prioritization among some defaults, i.e., a chain, there may be other defaults that have incomparable priority relative to these. E.g., there may be several chains (columns) in parallel. In this case, protection conditions to account for the "incomparables" are needed even when decomposing (serially or conjunctively) within any one chain. Interestingly, however, what Theorem 5.35 (serial, any sequence, general) tells us is that the canonical protection tactic then entirely eliminate s the sequence-dependence of the serial decomposition. (Recall that propositionality implies the well-foundedness pre-condition for Theorem 5.35.)

Part of the advantage of our notion of hierarchical decomposition based on modules, however, is that one can exploit **totality within a module**. That is, Theorem 5.69 can often be applied to decompose further within a module's phase or slice, even when the global axiom set does not fulfill the preconditions of the Theorem. E.g., if there is a chain (column) of propositional defaults.

5.13.3 For-Sure Beliefs As Highest-Priority Defaults

Observation 5.74 (Base Vs. Default)
Our next result is, mainly, a consequence of the 2-ary strict-priority case of serial decomposition along prioritization (Corollary 5.61). It shows that, in circumscription, one can view for-sure premise beliefs as simply a special case of default premise beliefs. I.e.:

> *Base axioms are, essentially, a special case of (prioritized) default axioms.*

In particular, then:

> *Updating with base axioms is, essentially, a special case of updating with (prioritized) default axioms.*

This supports our effort to formalize (prioritized) defaults as axioms in circumscription. It reinforces our interest in the problem of updating with new defaults. This perspective will be borne out

[7] More precisely, only guarding of them is required: see Observation 5.40.

further in chapter 6. There, when investigating some special-case localities (disjoint sub-languages in section 6.2 and sympathetic-solitary in section 6.5), we will discover safeties of updating that are essentially the same for base updates as for default updates.

Note that in practical default reasoning, it is usual to assume that the for-sure beliefs are consistent with each other, i.e., collectively satisfiable.

Theorem 5.75 (For-Sure Beliefs As Highest-Priority Defaults)
In CLD without fixture axioms or fixed function symbols (see discussion of this subtlety below): When satisfiable, any group E of base axioms is equivalent to a highest-priority group of default axioms, one per base axiom in the group.

More precisely:
Let BB be the conjunction of the rest of the base axioms (if there are any such; if not, let $BB[Z]$ be $True$).
Suppose that E and BB are collectively satisfiable, or else that BB without E is unsatisfiable:

$$\exists Z.\, BB[Z] \;\models\; \exists Z.\;\; BB[Z] \wedge \textstyle\bigwedge_{i \in N} Ei[Z] \tag{0}$$

where N is the index set of the tuple E.
Then

$$PDC(BB \wedge (\textstyle\bigwedge_{i \in N} Ei[Z]); D; R; Z) \;\equiv\; PDC(BB; E, D; R2; Z) \tag{G}$$

Here, on the right-hand-side of (G): the defaults E form one module $(E; RI)$, the defaults D form another module $(D; R)$, and the global pre-order $(D, E; R2)$ is the result of prioritizing the E module at higher priority than the D module. Thus all of the defaults E have higher priority than all of the other defaults D. The prioritization RI internal to the higher-priority (E) module may be anything: it is irrelevant; e.g., it may be empty.

Note that the pre-conditions of this theorem *prohibit fixtures*. The reason is that fixture might block the defaults E from going through. This prohibition *includes function* symbols; this is an exception to our **assumption** (page 36) that all functions are fixed. This requirement is not terribly onerous: see Theorem 2.24. The **non-superfluity** of this prohibition on fixture is shown by the counterexamples in Observation 5.70. See also that Observation for more discussion of prohibition on fixture, including of functions.

Note that one can read this equivalence result in the other direction too: a collectively satisfiable, highest-priority group of closed defaults is just equivalent to their for-sure conjunction.

Proof Overview: The essence is: 1) Collective satisfiability means that the defaults E do not conflict with each other, and thus go through: we formalize this as a lemma. 2) Highest-priority means that they do not have to worry about conflicting with any of the other defaults. We combine serial decomposition for totally-prioritized groups, Corollary 5.61, with the lemma. See Appendix.
□

Observation 5.76 (Base As Higher-Priority Than Fixtures Too)
One can view the prohibition on global fixture in Theorem 5.75 as saying, in effect, that (collectively satisfiable) base axioms correspond to defaults that have higher priority than all of the fixtures as well as all of the other defaults. This makes sense when one remembers that: in section 3.5.4, we showed that any fixture is equivalent to pair of polar defaults, and that CLD / PDC without explicit fixture is expressively equivalent to CLD / PDC with fixture.

Chapter 6

More Localities: Orthogonality and Sympathy

In this chapter, we develop more results about strong localities available in special cases. In section 5.13, we studied strong localities for special cases defined primarily in terms of prioritization. Here we investigate several special cases defined primarily in terms other than prioritization; our results are interesting even for parallel (i.e., empty) priority. For these special cases, as with total prioritization, we show that defaults are guaranteed not to conflict at all: we develop clean conjunctive decompositions, and strong safeties of updating; some are at coarse grain size and some are at fine grain size. And as with total prioritization, we are able to show that one case collapses to first-order with opportunities for new inference algorithms.

We observe that the bases for locality structure available in these cases falls into two categories: what we call "orthogonality" and "sympathy". The concept of logical orthogonality corresponds to the intuition of passing by on different geometric planes, or of "ships passing in the night". More formally, orthogonality is a kind of logical independence. The concept of sympathy corresponds to the intuition of pulling in the same direction, or working in harmony. More formally, we define directionality in terms of syntactic positivity and negativity. By contrast, with total prioritization, the concept of priority dominance corresponded to the intuition of winning in a battle.

Guide to Reader: Section 6.1 provides a summary / overview of the whole chapter.

6.1 Introduction and Summary

Strategic Overview:
In this chapter, we develop more results about strong localities available in special cases.

In section 5.13, we developed strong locality results for special cases that are defined primarily in terms of prioritization. We showed that prioritization was a basis for the structure of locality in an axiom set / theory. In particular, we showed that *clean* conjunctive and serial decomposition is available, at *coarsest* grain, for *part* of the theory when there was total prioritization between modules; and that clean serial decomposition is available, at *finest* grain, for the *whole* theory when those modules consisted of propositional defaults. In the latter case, we were able to show a *collapse* to first-order for each phase, and give a new inference *algorithm*. These clean decompositions also implied *safeties* of updating.

In this chapter we investigate several **special cases defined primarily in terms other than prioritization**. For these cases, we show that defaults are guaranteed not to conflict at all. For all of these cases, we develop **clean conjunctive decompositions, and strong safeties of updating**. However, these cases **differ in the grain size of non-conflict**, e.g., in the grain size of clean slices: group (of defaults) versus group, group versus one, and one versus one. In addition, for one special case, we are able to show collapse to first-order with opportunities for new inference algorithms.

Our results for these special cases are interesting even for globally parallel (i.e., empty) priority. Absence of conflict between groups makes the prioritization between those groups irrelevant.

Special-Case Builds on General-Case: Our special-case locality results in this chapter, as in section 5.13, build, in two ways, on our general-case canonical decomposition results and methods in chapter 5. Firstly, we use the general-case decompositions **to help prove** the special-case results about decomposition. And we use our general method of **differential decomposition** (Observation 4.6) together with those special-case decompositions, to prove special-case safeties of updating. Secondly, general-case decomposition may result in slices or phases that fall into one or more different special-case classes. Thus the general-case decompositions tell us how to **glue together** special-case results, using them for parts, rather than all, of a global axiom set / theory.

Synergy in Combining Special Cases; Running Example Another theme running through this chapter is the synergy available via our overall strategy of **hierarchical decomposition**. We show that special case results can be combined within our general approach, to achieve synergy. Decompositions of several different kinds can be combined for one global axiom set, to achieve greater locality and safeties than are available from any one kind for that axiom set. We demonstrate this with a large-ish running example, in the domain of common-sense reasoning, that builds as the chapter progresses. We show that **this example combines, and exploits, all of our main special cases in this thesis**: all of the different ones in this chapter, plus the total-prioritization special cases in section 5.13.

A key idea is that one special case may be used to decompose at coarser grain, and then another special case may be used to decompose at finer grain. Another key idea is that, for a given update, one special case may be used to guarantee safety of one part of the previous theory, while another special case is used to guarantee the safety of a second part of the previous theory.

Bases For Locality Structure: Orthogonality and Sympathy:

We observe that the bases for locality structure available in these special cases fall into two categories: what we call "orthogonality" and "sympathy".

Concept of Orthogonality:
In chapter 5, and especially in section 5.13 (totality), the concept of priority dominance corresponded to the intuition of winning in a battle.

The concept of logical orthogonality corresponds to the intuition of passing by on different geometric planes, or of "ships passing in the night". More formally, orthogonality is a kind of logical independence. The standard notion of logical independence (see, for example, [Enderton, 1972] page 38) is that a set of axioms is independent if no one member is tautologically implied by the remainder of the set. In orthogonality, we apply the idea of independence to sets of groups of formulas and axioms, not just to sets of individual axioms. The basic idea is that there is no influence on a whole group by the rest of the groups; by influence we mean that some of the members of the group are implied. Elsewhere, we show how to define orthogonality precisely: in terms of factorability of models (omitted to save space and increase focus). But for our purposes in this thesis, the informal concept suffices, since we are concentrating on a couple of specific kinds of it.

Concept of Sympathy:
The concept of *sympathy* corresponds to the intuition of pulling in the same direction, or working in harmony. Technically, we use the ideas of syntactic positivity and negativity to capture the notion of directionality. There exist several previous results on positivity and negativity in circumscription (which to us, suggested this as a fruitful line of approach). The basic idea of sympathy is that some formulas are sympathetic with each other if their mention of predicates occurs with the same sign. More generally, the role of predicates can be generalized to (elementary) formulas. As with orthogonality, for reasons of space and focus, since we are concentrating on a couple of specific kinds of it, we leave the concept of sympathy informal.

Disjoint Sub-Languages:
In section 6.2, we investigate a very basic case of orthogonality: disjointess of mentioned predicate (and function) symbols.
Terminology: we call this **disjoint sub-languages** (of the base language $\mathcal{L}$).
We show that when the set of base and default axioms is partitionable by disjoint groups of mentioned symbols, then the theory is partitionable on this basis also: it is cleanly slice-able into sub-theories, one per partition. I.e., there is no conflict at all between partitions, only within them.

We discover non-trivial subtleties along the way, however. Prioritization, especially non-layered prioritization, and fixture each complicate matters.

We show that the clean conjunctive decomposition available for the disjoint sub-languages case implies:
1) perfect locality of both forward and backward inference: each partition slice is sound and complete for its sub-language; and 2) powerful safeties of updating: all partitions' conclusions are safe

whose sub-languages are not mentioned by an update.

Most large axiom sets of interest for applications, whether in monotonic logic or non-monotonic logic, do not display much perfect partitionability of symbols. Hence, one may wonder: why care much about the disjoint sub-languages case? Our answer is that the real power of our disjoint sub-languages results comes when they are combined with a kind of **monotonic**-logical equivalence-preserving "definitional" **reformulation**. Then a much broader class of **disjointly describable** axiom sets can be shown to have clean slicings. In section 6.3, we give a small example of disjoint describability. Elsewhere [Grosof, 1992c] we have detailed this approach and shown its fruitfulness: the research reported there was built on, and chronologically came after, the work reported in this thesis.

The non-conflict in the disjoint sub-languages case, and in the disjointly describable case, is coarse grain: it is between groups. Disjoint sub-languages (or describability) is a restriction on the entire axiom set.

Conservative Extension:
In section 6.4, we investigate a second very basic case of orthogonality: conservative extension. Conservative extension is a standard logical notion (see Definition 3.6). We show that any base or default update is globally monotonic when its formula conservatively extends the previous base. Moreover, the theory after the update has a simple clean conjunctive or serial decomposition: one slice or phase for the update, and one for the previous (rest of the) axiom set. An example of a conservatively extending update is: to add a new rule (i.e., clause) that mentions a new predicate p.

The non-conflict in the conservative extension case has mixed grain: it is between one (the update) and a group (the previous axioms). Note that conservative extension is a restriction on the update, relative to (i.e., applicable to) *any* previous axiom set.

Sympathetically Solitary:
In section 6.5, we find a non-conflicting case of sympathy that generalizes Lifschitz' "solitary" class. In this case, we are able to give a first-order "explicit solution" to the entire circumscription. By "explicit solution", we mean a first-order-logical axiomatization of the non-monotonic conclusions plus the monotonic conclusions; from the "explicit solution", the entire non-monotonic theory can be concluded using only monotonic first-order-logical inference. This solution is computable in time that is polynomial in the size of the global CLD axiom set. This implies the opportunity to develop tractable and decidable inference and belief revision algorithms for further specializations of this "sympathetically solitary" class.

The explicit solution also implies strong (global) safeties of updating.

The non-conflict in the sympathetically solitary case is at finest-grain: the explicit solution corresponds to a **finest-grain clean** conjunctive decomposition with one default (or default instance, essentially) per slice.

Sympathetically solitary is a restriction on the entire axiom set. Perhaps the most important aspect of this restriction is that default rules can not be chained: the default reasoning is "shallow".

Strong Sympathy:
In section 6.6, we show that another non-conflicting case of sympathy is when new base or default formulas are formed purely and positively from previous default and fixed formulas. Such "strongly sympathetic" updates are globally and forever (Definition 4.17) monotonic.

We show that this result is especially applicable when it is combined with another kind of monotonic-logical equivalence-preserving reformulation: **equivalence given the base**.

The non-conflict in the strong sympathy case has mixed grain: it is between one (the update) and a group (the previous axioms). Strong sympathy is a restriction on the update, relative to (i.e., applicable to) *any* previous axiom set that is *parallel*. (Elsewhere, we generalize this to permit prioritization.)

Advantage of Defaults and CLD Over Minimizing Predicates:
Many of our results in this chapter show the advantage of thinking about updates in terms of maximizing *default* formulas and CLD, not just minimizing predicates. In particular, several of our results show the similarity of default updates to base updates. This similarity is manifested more clearly when described in terms of maximized defaults and CLD, than in terms of minimizing predicates, i.e., in terms of abnormality theories. Usually, the abnormality-theory version is more complicated to state and less illuminating. The above similarity also accords with our intuition that base axioms and updates can be viewed as a special case of default axioms and updates, which was formalized in sub-section 5.13.3.

Lack of Conflict Makes Prioritization Irrelevant:
Running through the results in this chapter is the phenomenon that non-conflict between groups (or individual) defaults makes prioritization between those groups irrelevant.

6.2 Disjoint Sub-Languages

Overview of Main Result about Decomposition:
In this section, we investigate a very basic case of orthogonality: disjointess of mentioned predicate (and function) symbols.
Terminology: we call this **disjoint sub-languages** (of the base language $\mathcal{L}$).

More precisely: In a global prioritized default circumscription, suppose the base and default formulas form k partitions, where there is no overlap in the mentioned symbols between partitions. Then, intuitively, the base and defaults in one partition "have nothing to do with" those in any other. Disjointness of mentioned symbols means that the base and defaults are not "connected" syntactically in the base language, between partition groups. I.e., there is "no syntactic mixing" between the partitions.

In this section, we show that, indeed, when the set of base and default axioms is so partitionable by disjoint groups of mentioned symbols, then *often* the theory is partitionable on this basis also: the global PDC is conjunctively decomposable into k clean slices, one per partition. I.e., there is no conflict at all between partitions, only within them.

Previous Work; Non-Triviality:

Amazingly enough, this disjoint sub-languages case has never been previously addressed in the circumscription literature (and not much elsewhere in the NMR literature), perhaps because it was thought of as trivial. Indeed, in Default Logic, for example, it is fairly trivial, due to the construction of its definition. By contrast, we discover that, in circumscription, at least, the "disjoint sub-languages" case is not trivial to prove: prioritization, especially non-layered prioritization, creates subtlety and difficulty. In the prioritized case of (predicate or default) circumscription, the minimization condition "mixes up" the different predicates (and functions), in such a way that, on the surface, it is not immediately apparent that the theory is partitionable (i.e., cleanly slice-able by partition). We also discover another subtlety: the role of fixture.

Prioritization's Role; Conditions on It; May Overlap:

In CLD, suppose one views prioritization as part of the "language" and its syntax. Then the natural ramification of "no syntactic mixing" between partitions is to require that the prioritization between defaults in different partitions be empty. I.e., it is appropriate to require that the partitions correspond to modules (Definition 2.62), with parallel prioritization between those modules.

We show that this condition indeed suffices for the clean conjunctive decomposition.

But we show also that this condition on prioritization can be generalized to permit overlap between the partitions. Firstly, there may be strict priority between partitions, as long as it is uniform across each whole partition. I.e., if the partitions correspond to modules, then there may be arbitrary prioritization between the modules. In this case, we say that the prioritization is "modular by partition". Secondly, the prioritization may be globally layered, even if it is not modular by partition. Thirdly, if the prioritization between the partitions can be *updated* to be modular or globally layered, that suffices too.

Fixture's Role; Conditions On It; May Overlap:

In CLD, suppose one views fixture as part of the "language" and its syntax, as with prioritization. Then the natural ramification of the "no syntactic mixing" between partitions is to require that the symbols mentioned in the fixture (axioms') formulas should not overlap between the different partitions. This view is supported, for example, by the equivalence of a fixture to a pair of polar defaults (Lemma 3.42).

However, we show that *any* (globally) fixed symbol *can* overlap between partitions.

Terminology: Partition: In the following discussion (and in the rest of this thesis), we will still speak of this as a partition of the axioms according to partitionability of the (non-fixed) language into sub-languages. That is, we will employ the term "partition" in a slightly loose sense, where fixed symbols are permitted to overlap between the sub-language "partitions".

Theorem 6.1 (Clean Decomposition, Disjoint Sub-Languages)
In a global $PDC(B; D^N; R; fix\ W; Z)$ where the only explicit fixing is of symbols (W): Let $Z \stackrel{\text{def}}{=} \langle Y, W \rangle$.
Suppose that $\{B1[Y1, W], \ldots, Bk[Yk, W]\}$ is a partition of the base axioms $B[Y, W]$, and that $\{D1[Y1, W], \ldots, Dk[Yk, W]\}$ is a partition of the default formulas $D[Y, W]$, where the predicate tuples $Y1, \ldots, Yk$ are a (disjoint) partition of Y. I.e., in terms of CLD, let there be a partition, of the base and default axioms, where the sub-languages used in each element of the partition are disjoint except possibly for fixed symbols. [1]
If the condition (0) (see below) on the prioritization is satisfied, then

$$PDC(B; D; R; fix\, W; Y, W) \quad \equiv \quad \bigwedge_{j=1}^{k} PDC(Bj; Dj; RIj; fix\, W; Y, W)$$

where Nj stands for the the index tuple, and $RIj \stackrel{\text{def}}{=} R^{Nj}$ stands for the internal prioritization, of the group of defaults Dj.
Note that the Y on the right-hand side above can, equivalently, be replaced by Yj, since, for each j:

$$PDC(B; Dj; RIj; fix\, W; Y, W) \quad \equiv \quad PDC(B; Dj; RIj; fix\, W; Yj, W)$$

The condition (0) on the prioritization R is defined as follows.
Either (0.1) R is the composition (Definition 2.51 and Theorem 2.56) of some external prioritization RE with the tuple RI of the internal prioritizations of each partition group of defaults (in this case we **define** R to be **modular** by partition) [2] ;
or, (0.2) R is layered;
or, more generally, (0.3) R has an "upper bound" RM that coincides with R on the partition groups and satisfies either (0.1) or (0.2).
Here, by bounding we mean that:
$R \leq RM$
and by coinciding we mean that:
$\forall j = 1, \ldots, k.\ R^{Nj} = RM^{Nj}$

Proof Overview: Surprisingly complicated. Uses canonical conjunctive decomposition (section 5.4): both hierarchical by modules, and one-by-one. Also uses the monotonicity of prioritization (Theorem 2.58).

[1] If one relaxes the **assumption** (page 36) that all function symbols are fixed, then the overlap between partitions may include only the function (and predicate) symbols that are fixed. In that case, re-define Z,Y, and W to include function symbols as well as predicate symbols.

[2] Note that we are using partition in this section in a somewhat different sense than we did in section 5.13, where we spoke of partitioning the prioritization itself into 2 strictly-prioritized modules.

The essence is to use the ability to separate existential quantifiers in the augmentation part of the circumscription. Non-layered prioritization makes this tricky: hence the prioritization conditions in the theorem. See Appendix. □

Parallel Case Is Covered:
Note that the special case of empty (i.e., parallel) global prioritization satisfies (0). (The proof is much easier for the parallel case.) Thus the theorem yields clean slicings even for default reasoning without priorities.

Lack of Counterexample:
We suspect that the result may hold without constraint on the global prioritization R: we have not yet found a counterexample that shows the non-superfluity of condition (0). That condition is used to make the proof go through.

Irrelevance; Locality; Forward Inference; Selectiveness:
In terms of the discussion in sub-section 4.5.3: the clean-ness of the conjunctive decomposition in the theorem tells us that
forward inference can be performed locally (with soundness), selectively, and concurrently: by partition. The NM aspect of inference is kept intra-partition. In performing inference within each partition slice, the axioms in the other partitions are irrelevant (except for the fixtures).

Theorem 6.1 also immediately yields an even stronger result about inference than is provided by the clean-ness of the slicing. Each partition slice is *complete*, as well as sound, for inference on its corresponding sub-language of the base language $\mathcal{L}$, when there are no overlapping function symbols.

Corollary 6.2 (Local Completeness of Inference, Disjoint Sub-Lang.)
In Theorem 6.1, suppose also that the fixed symbols too (including function symbols as well as predicate symbols) are partitionable in the same manner as the base and default axioms. Suppose furthermore that the the global circumscription is satisfiable. (If it is not, then inference is pretty pointless anyway.)
Then the j^{th} slice is sound and complete, relative to the global theory, for inference over its corresponding sub-language. That sub-language consists of the formulas that mention only the symbols $\langle Yj, Wj\rangle$. In other words, that slice entails such a formula if and only if it is entailed by the global theory. Note that this local soundness and completeness holds both for backward inference, e.g., query-answering, and for forward inference.

More generally, to perform inference using any subset Y of the symbols Z, one need only work in the conjunctive combination of those slices whose predicates cover that subset Y.

Proof Overview: Short. Uses existential projection applied to Theorem 6.1. See Appendix. □

Necessity for Partitioning the Fixed Symbols:

The reason that the fixed symbols need to be partitioned too in order to guarantee local completeness is easily shown:
Otherwise, a slice may lack the axioms, contained in some other slice, that entail the truth of some formula purely in fixed symbols.

Theorem 6.1 also immediately yields a powerful result about belief revision.

Corollary 6.3 (Safety of Updating, given Disjoint Sub-Languages)
In CLD: Suppose that the previous axiom set $\mathcal{A}$ meets the conditions of Theorem 6.1. Let J stand for the (index) set of the previous partitions, according to some particular application of that Theorem. (For a given axiom set, there may be more than one decomposition according to Theorem 6.1.) Suppose that the update $\mathcal{U}$ consists of base, default, and prioritization axioms (not fixture axioms), such that the formula parts of the base and default axioms in $\mathcal{U}$ mention only predicates from a (possibly empty) proper subset S of the previous partitions J, plus fixed predicates.

Then all of the previous conclusions in (the conjunctive combination of) the rest of the previous partitions' (i.e., $J - S$'s) slices are safe under the update, if an additional prioritization condition (1) is met.

(1): after the update, the new axiom set $\mathcal{A}\&\mathcal{U}$ also meets the prioritization condition (0) of Theorem 6.1 for the partition set $\{US\} \cup (J - S)$, where US is the partition element corresponding to the merging of the update with the previous partition elements S.

Condition (1) is met, for example, when $\mathcal{A}$ satisfies modularity cf. (0.1) in Theorem 6.1 and the group of defaults in the update all have the same prioritization relative to all the (previous) defaults in the partition elements $J - S$. Condition (1) is also met, for example, when the prioritization before and after the update is (globally) layered.

Note that the update is globally monotonic if it mentions no previous non-fixed predicates.

Proof : Use Theorem 6.1. The guiding spirit is Observation 4.6: differential decomposition. Before the update, the CLD theory is cleanly decomposable into one slice for each member of J. After the update, the CLD theory is cleanly decomposable into one slice for each member of $J - S$, plus one slice corresponding to US. But the slices corresponding to $J - S$ are unchanged from that previous decomposition, except possibly for some added prioritization. By the monotonicity of prioritization (Theorem 2.58), each of those new $J - S$ slices implies the corresponding previous $J - S$ slice. $\Box$

Example 6.4 (Meetings and Animals: Disjoint Sub-Languages)
In section 5.12, we gave a large example about Meetings which illustrated total prioritization over modules (sub-section 5.13.1). In this chapter, we continue that example, and expand on it progressively to show how to combine all of our main special-cases of non-conflict.

We start from Example 5.59 before the update (with the conferee default) there. Again, throughout this chapter, as a shorthand for conjunctions of for-sure assertions of positive or negative literals, we list the satisfying objects, or, more generally, tuples. In this context, we use "..." to indicate that there are additional satisfying tuples not shown explicitly; for simplicity's sake, we assume these objects are distinct from all other explicitly-shown objects. As usual in our examples, we include the uniqueness of all names. And, in addition, here we include a domain closure axiom: say, with 0-ary functions only, and with the domain element (0-ary function) symbols being the first googolplex ($10^{10^{100}}$) ASCII strings. Consider the uniqueness of names and the domain closure axioms to be implicitly included in the global base below. (Technically, this makes the slices below not quite clean: those (base) axioms need to be included in each slice. But we'll ignore that fact below, as we usually do throughout this thesis, in order to save breath.)

Suppose one updates that global axiom set (let us call it $\mathcal{U}_1$) with the following axioms (let us call them $\mathcal{U}_2$) about Animals:

$\bullet\!\!>\ \forall x.\, bat(x) \supset mammal(x)$

$\bullet\!\!>\ \forall x.\, dog(x) \supset mammal(x)$

$\bullet\!\!>\ +bat:\ \ Betsy, Joe, June, Jackie, Drac, Yum, \ldots$

$\bullet\!\!>\ +dog:\ \ Fido, Spot, Siccem, Jumper, Ruff, Wag, Heywood, \ldots$

$(a1) :>\ mammal(x) \supset 4legs(x)$

$(a2) :>\ bat(x) \supset 2legs(x)$

$\mathcal{PREFER}(a2, a1)$

$\bullet\!\!>\ \forall x.\, \neg(2legs(x) \wedge 4legs(x))$

$\bullet\!\!>\ \neg mammal:\ \ Allie, Godzilla, Choochoo, \ldots$

$\bullet\!\!>\ +mammal:\ \ Fawn, Sharon, Amy, Bozo, \ldots$

$\bullet\!\!>\ \neg 2legs:\ \ Fido, June, \ldots$

$\bullet\!\!>\ \neg 4legs:\ \ June, Wag, Starcrossed, \ldots$

Then the global axiom set after this update has two disjoint sub-languages, one about Meetings and another about Animals, and meets the conditions in Theorem 6.1. By Theorem 6.1, therefore, there is a clean conjunctive decomposition into two slices: one for Meetings, and one for Animals. The one for Meetings contains all of the previous axioms, i.e., all of the axioms $\mathcal{U}_1$ in Example 5.59. We call these the Meetings axioms. The one for Animals contains the rest, i.e., those ($\mathcal{U}_2$) listed immediately above. We call these the Animals axioms. By Corollary 6.3, therefore, the update is globally monotonic. Among the many sanctioned NM conclusions after this update is:

$4legs(Sharon)$

Next, suppose there is another update $\mathcal{U}_3$:

$(a3) :>\ arachnid(x) \supset 8legs(x)$

$\bullet\!\!>\ arachnid:\ \ Lobo, Pete, Spad, \ldots$

$\bullet\!\!>\ \neg 8legs:\ \ Spad, \ldots$

Then this update, about Arachnids, forms a third sub-language partition, and a new clean slice. It is also globally monotonic. Thus, for example, $4legs(Sharon)$ is still a sanctioned NM conclusion after this update.

Next, suppose there is another update $\mathcal{U}_4$:

- $\bullet\!> \ \forall x.\, \neg(2legs(x) \wedge 8legs(x))$
- $\bullet\!> \ \forall x.\, \neg(4legs(x) \wedge 8legs(x))$
- $\bullet\!> \ \neg 4legs(Sharon)$

Then this update syntactically connects ("mixes") the Arachnid group of axioms to the Animals group of axioms. It results in two clean disjoint-sub-language slices after the update: the Arachnid group is **merged** with the Animals group. The update is not globally monotonic: the previous conclusion $4legs(Sharon)$ is retracted, for example. But Corollary 6.3 guarantees partial monotonicity: the safety of all the conclusions in the Meetings slice.

Next, suppose there is another update $\mathcal{U}_5$, consisting of the default update about conferees in Example 5.59. This just alters the Meetings group. Thus Corollary 6.3 guarantees the safety of all the previous conclusions in the Animals slice. Moreover, Theorem 5.55 guarantees the safety of all the previous conclusions in the Meetings slice that were derived solely from the (Meetings) defaults with higher priority than the new conferee default. Corollary 5.61 tells us this set of higher-priority previous Meetings conclusions is exactly the clean slice corresponding to the emergencies module. Thus all of the conclusions in (the conjunctive combination of) the emergencies slice *and* the Animals slice are safe. This illustrates the **synergy** created by combining two different kinds of special-case decompositions, to get more safeties and finer clean decomposition than is available from either alone. We are using one special case (total prioritization between modules) to decompose more finely, i.e., hierarchically, within another special case (disjoint sub-languages).

Next, suppose there is another update $\mathcal{U}_6$, about Feelings:

- $\bullet\!> \ \forall x, y.\, \neg(loves(x,y) \wedge hates(x,y))$
- $\bullet\!> \ \forall x, y.\, \neg[(loves(x,y) \vee hates(x,y)) \wedge indifferent(x,y)]$

$(f1) :> \ loves(Amy, Bozo)$

that includes a revisable belief that Amy loves Bozo. This forms a new disjoint-sub-language partition: the Feelings group. It is thus globally monotonic as an update.

Next, suppose there is another update $\mathcal{U}_7$:

$(f2) :> \ hates(Amy, Bozo)$

$\mathcal{PREFER}(f2, f1)$

, i.e., fresher information that Amy has changed her heart about Bozo; followed by another update $\mathcal{U}_8$:

$(f3) :> \ indifferent(Amy, Bozo)$

$\mathcal{PREFER}(f3, f2)$

, i.e., yet fresher information that Amy has given up on Bozo and developed a stony heart.

Each of these last two updates alters only the Feelings group. Thus, after each, all conclusions in (the conjunctive combination of) the Meetings and Animals slices are safe. Moreover, the Feelings slice is total-propositional: Theorem 5.69 tells us that, after each of the updates $\mathcal{U}_6$, $\mathcal{U}_7$, and $\mathcal{U}_8$, we can exhaustively re-compute the Feelings slice's conclusions (more precisely, its generating axioms; recall the discussion of generating axioms in sub-section 4.3.1., and see discussion of generating axioms in section 6.5, especially point 2. of Observation 6.13) decidably using the serial consistency-checking algorithm (Theorem 5.69). This again illustrates the **synergy** created by combining **three different kinds of special-case decompositions**, to get more locality, e.g., clean decomposition, than is available from any one alone. Here, we are using two *different* special cases (total propositional for Feelings, and total prioritization over modules for Meetings) to decompose more finely, i.e., hierarchically, within another special case (disjoint sub-languages).

6.3 Disjoint Describability; Reformulation

Most large axiom sets of interest for applications, whether in monotonic logic or non-monotonic logic, do not display much degree of the almost perfect partitionability, by mentioning of (especially predicate) symbols, required to meet the disjoint-sub-languages case cf. Theorem 6.1. Hence, one may wonder: why care much about the disjoint sub-languages case? Our answer is that the real power of our disjoint sub-languages results comes when it is combined with a kind of **monotonic**-logical equivalence-preserving "definitional" **reformulation**. Then a much broader class of **disjointly describable** axiom sets can be shown to have clean slicings.

In this section, we give the concept, and a few examples, of disjoint describability. Elsewhere [Grosof, 1992c] we have detailed this approach and shown its fruitfulness: the research reported there was built on, and chronologically came after, the work reported in this thesis.

Concept of Disjoint Describability:

The essential idea of disjoint describability is:

syntactic partitionability after definitional reformulation.

More precisely: Suppose one definitionally reformulates an (original) axiom set into an alternative representation: a new axiom set, expressed in new predicate and function symbols, that is, essentially, equivalent to the original. And suppose this new axiom set is partitionable into disjoint sub-languages cf. Theorem 6.1. Thus the new axiom set is equivalent to the conjunctive composition of some partition slices in the new reprentation. Finally, suppose that these partition slices can be reformulated, essentially, equivalently back into the original representation.

Then the original axiom set is equivalent to the conjunctive composition of these partition slices in the original representation.

...Is Complicated:

Of course, there are many issues and subtleties involved in formalizing disjoint describability. One constellation revolves around definitional reformulation, including: the notions of essential equivalence, the mappings between representations, their invertibility, necessary and sufficient preconditions on the original axiom set to permit such, etc.. Another constellation revolves around identifying NM-logical special cases with strong locality via disjoint describability. A third constellation revolves around how to automate the practical recognition and exploitation of such disjointly describable for useful applications. In short, disjoint describability could easily be the subject of a whole other dissertation.

Individual-Wise Clean Slicing:
Next, we focus on one special case of disjoint describability: decomposing "individual-wise". Here, the basic idea is to view each atom in the base language $\mathcal{L}$ as definitionally reformulable, in an alternative representation, into a 0-ary predicate, i.e., a proposition.

Example 6.5 (Animals, Individual-Wise Locality, via Disjoint Describ.)
Continuing Example 6.4:
Next, suppose there is another update $\mathcal{U}_9$:

$\bullet\!>\ \neg 2legs(Joe)\ \wedge\ \neg 4legs(Joe)$

$\bullet\!>\ \neg 2legs(Spot)\ \wedge\ \neg 4legs(Spot)$

This update affects only the Animals slice. (It thus leaves the Meetings and Feelings slices safe.) It causes some retractions within the Animals slice: the previous default conclusions

$2legs(Joe)$

$4legs(Spot)$

are violated by the new for-sure information.

Our disjoint sub-languages results can guarantee nothing about the safety of any of the previous conclusions in the Animals slice: those results, in effect, can only say: "all bets are off".

Using disjoint describability, however, it is possible to show a very powerful safety of updating for this last update: that *all* of the conclusions (in the Animals slice) about individuals other *Joe* and *Spot* are safe. I.e., all conclusions about individuals other than those mentioned in the update are safe.

How? We illustrate with a simplified version of the above example.

Example 6.6 (Bats, Individual-Wise Locality, via Disjoint Describ.)
Consider, as a global axiom set, just the following axioms from the Animals group in Example 6.4:

$\bullet\!>\ \forall x.\, bat(x) \supset mammal(x)$

$:>\ bat(x) \supset 2legs(x)$

$\bullet\!>\ bat(Betsy)$

$\bullet\!>\ bat(Drac)$

$\bullet\!>\ bat(June)$

$\bullet\!>\ \neg 2legs(June)$

We introduce a new propositional (0-ary) predicate for each ground atom that is formable from the symbols appearing in this global axiom set. We define the definitional reformulation mapping between representations via a meta-theoretical "definitional reformulator". The definitional reformulator consists of the monotonic-logical axioms:

$nbatBetsy\ \stackrel{\text{def}}{\equiv}\ bat(Betsy)$

$nbatDrac\ \stackrel{\text{def}}{\equiv}\ bat(Drac)$

$nbatJune\ \stackrel{\text{def}}{\equiv}\ bat(June)$

$\forall x.\, nbatRest(x)\ \stackrel{\text{def}}{\equiv}\ bat(x) \wedge (x \neq Betsy) \wedge (x \neq Drac) \wedge (x \neq June)$

$$nmammalBetsy \stackrel{\text{def}}{\equiv} mammal(Betsy)$$
$$nmammalDrac \stackrel{\text{def}}{\equiv} mammal(Drac)$$
$$nmammalJune \stackrel{\text{def}}{\equiv} mammal(June)$$
$$\forall x.\, nmammalRest(x) \stackrel{\text{def}}{\equiv}$$
$$mammal(x) \wedge (x \neq Betsy) \wedge (x \neq Drac) \wedge (x \neq June)$$
$$n2legsBetsy \stackrel{\text{def}}{\equiv} 2legs(Betsy)$$
$$n2legsDrac \stackrel{\text{def}}{\equiv} 2legs(Drac)$$
$$n2legsJune \stackrel{\text{def}}{\equiv} 2legs(June)$$
$$\forall x.\, n2legsRest(x) \stackrel{\text{def}}{\equiv} 2legs(x) \wedge (x \neq Betsy) \wedge (x \neq Drac) \wedge (x \neq June)$$

Using these new predicate symbols and the definitional reformulator, we definitionally reformulate the NM axiom set into a new axiom set:

$\bullet\!>$ $nbatBetsy \supset nmammalBetsy$
$\bullet\!>$ $nbatDrac \supset nmammalDrac$
$\bullet\!>$ $nbatJune \supset nmammalJune$
$\bullet\!>$ $\forall x.\, nbatRest(x) \supset nmammalRest(x)$
:> $nbatBetsy \supset n2legsBetsy$
:> $nbatDrac \supset n2legsDrac$
:> $nbatJune \supset n2legsJune$
:> $nbatRest(x) \supset n2legsRest(x)$
$\bullet\!>$ $nbatBetsy$
$\bullet\!>$ $nbatDrac$
$\bullet\!>$ $nbatJune$
$\bullet\!>$ $\neg n2legsJune$

This new representation has four disjoint-sublanguage partitions cf. Theorem 6.1: one for each named individual, plus one for the remainder ("Rest") case. Thus the new axiom set has four **clean slices**.

The first, about Betsy, is:

$\bullet\!>$ $nbatBetsy \supset nmammalBetsy$
:> $nbatBetsy \supset n2legsBetsy$
$\bullet\!>$ $nbatBetsy$

The second, about Drac, is:

$\bullet\!>$ $nbatDrac \supset nmammalDrac$
:> $nbatDrac \supset n2legsDrac$
$\bullet\!>$ $nbatDrac$

The third, about June, is:

$\bullet\!>$ $nbatJune \supset nmammalJune$
:> $nbatJune \supset n2legsJune$
$\bullet\!>$ $nbatJune$
$\bullet\!>$ $\neg n2legsJune$

The fourth, about the Rest, is:

$\bullet\!>$ $\forall x.\, nbatRest(x) \supset nmammalRest(x)$

$:>$ $nbatRest(x) \supset n2legsRest(x)$

We then invert the reformulation mapping, again using the definitional reformulator, to map back to the original symbols.

The Betsy slice is mapped back to:

$\bullet\!>$ $bat(Betsy) \supset mammal(Betsy)$

$:>$ $bat(Betsy) \supset 2legs(Betsy)$

$\bullet\!>$ $bat(Betsy)$

The Drac slice is mapped back to:

$\bullet\!>$ $bat(Drac) \supset mammal(Drac)$

$:>$ $bat(Drac) \supset 2legs(Drac)$

$\bullet\!>$ $bat(Drac)$

The June slice is mapped back to:

$\bullet\!>$ $bat(June) \supset mammal(June)$

$:>$ $bat(June) \supset 2legs(June)$

$\bullet\!>$ $bat(June)$

$\bullet\!>$ $\neg n2legsJune$

The Rest slice is mapped back to:

$\bullet\!>$ $\forall x.\, [bat(x) \wedge (x \neq Betsy) \wedge (x \neq Drac) \wedge (x \neq June)] \supset mammal(x)$

$:>$ $\forall x.\, [bat(x) \wedge (x \neq Betsy) \wedge (x \neq Drac) \wedge (x \neq June)] \supset 2legs(x)$

Thus, we arrive at four **slices, without protection conditions**, in the original language. Their significance is that the *original* global axiom set is conjunctively decomposable into these four slices. There is one per individual, plus one for the Rest case. These slices are not exactly clean, since they are not precisely subsets of the original global axiom set: they also include instantiations of the original universal base axioms and of the original open default axioms. However, each of these slices, especially the three individual-wise slices, display a great degree of locality: more than if they were clean slices. Individual-wise is pretty fine grain. Therefore, inference can be done locally: e.g., about only one individual at a time.

This locality implies safeties as well, by applying our usual differential decomposition strategy (Observation 4.6). For example, suppose we update the original global axiom set with the base axiom

$\bullet\!>$ $\neg 2legs(Betsy)$

that violates the two-legged-ness default for Betsy. It can be shown, via the above reformulation method, that this update does not affect the slices for any of the other individuals. Thus all of the previous conclusions about the other individuals are safe under the update. For example, The Drac slice axiom set still entails the default conclusion that $2legs(Drac)$.

In this fashion, it can be shown that, in Example 6.5, the NM update about Joe and Spot leaves safe all of the previous conclusions about the other individuals.

6.4 Conservative Extension

Overview:

In this section, we investigate a second very basic case of orthogonality: conservative extension. Conservative extension is a standard logical notion (see Definition 3.6).

Definition 6.7 (Universal Closure)

By the *universal closure* of an open (or closed) formula $E[Z, x]$, we mean adding a universal quantifier for each free (individual / object) variable in E.

$\forall E \ \stackrel{\text{def}}{\equiv}\ \forall x.\, E[Z, x]$

E.g., if E is $governor(x, y, t) \supset governor(x, y, t + 1)$, then its universal closure is $\forall x, y, t.\, governor(x, y, t) \supset governor(x, y, t + 1)$. If E is closed, then it is its own universal closure.

Theorem 6.8 (Conservative Updates: Clean Decomposition and Safety)

Let $PDC(B; D; R; fix\ F; Z)$ be a previous global PDC.

We consider updating in terms of CLD.

Suppose the (elementary) **sentence** $E[Y, Z]$ conservatively extends (see Definition 3.6) the previous base $B[Z]$. Here Y are new symbols: they are distinct from Z.

Then E is **globally monotonic as a base update, or as a default update with any priority**. Moreover, in the case where E is a default, the default "goes through".

In more detail: the circumscription after the base update is equivalent to: the circumscription before the update, conjoined with E:

$PDC(B \wedge E; D; R; fix\ F; Y, Z) \ \equiv\ E[Y, Z] \wedge PDC(B; D; R; fix\ F; Z)$

Likewise, the circumscription after the default update, in the case when the default update does *not* include prioritization, is the same as in the case of a base update:

$PDC(B; D, E; R2; fix\ F; Y, Z) \ \equiv\ E[Y, Z] \wedge PDC(B; D; R; fix\ F; Z)$

where $R2^N = R^N$ and N indexes D; i.e., where the new prioritization agrees with the old on the previous defaults, and is empty elsewhere.

In the case where the default update *does* include prioritization, the circumscription after the update is at least that strong:

$PDC(B; D, E; R2; fix\ F; Y, Z) \ \supset\ E[Y, Z] \wedge PDC(B; D; R; fix\ F; Z)$

More generally, we consider an update containing **several defaults**, and permit those defaults to be **open**. Let E^{NU} be a tuple of open (or closed) *formulas* Ei. The conjunctive collection of the universal closures of these formulas is defined as:

$EC \ \stackrel{\text{def}}{\equiv}\ \bigwedge_{i \in NU} \forall Ei$

Suppose that EC conservatively extends B:

$B[Z] \ \models\ \exists Y.\, EC[Y, Z]$

Then an update consisting of one default axiom per Ei, with any prioritization, is globally monotonic. Moreover, each of the defaults "goes through". In the case where the update does *not* include prioritization, the circumscription after is equivalent to the circumscription before conjoined with their (universal closures') conjunction.

$$PDC(B; D, E; R2; fix\ F; Y, Z) \equiv EC[Y, Z] \wedge PDC(B; D; R; fix\ F; Z)$$

where $R2^N = R^N$ and N indexes D; i.e., where the new prioritization agrees with the old on the previous defaults, and is empty elsewhere.

In the case where the default update *does* include prioritization, the circumscription after the update is at least that strong:

$$PDC(B; D, E; R2; fix\ F; Y, Z) \supset E[Y, Z] \wedge PDC(B; D; R; fix\ F; Z)$$

Above, the cases without prioritization updating can be viewed as a **clean conjunctive decomposition** in which there are two slices. The first slice's axiom set is the entire previous axiom set. The second slice's axiom set is the update. Likewise, the above can be viewed as a **clean serial decomposition**, with the same constituent axiom sets, in either sequence.

The non-conflict in this conservative extension case has mixed grain: it is between one or a few (the update) and a larger group (the previous axioms). Note that conservative extension is a restriction on the update, relative to (i.e., applicable to) *any* previous axiom set.

Proof Overview: For the default case, we use canonical conjunctive decomposition, one-by-one: Corollary 5.21. The case with prioritization follows from the case without prioritization: by the monotonicity of prioritization, Theorem 2.58. New prioritization axioms may affect the prioritization among the previous defaults: by transitivity "through" a new default (recall discussion of such effects in the proof of Theorem 5.55). See Appendix for details. □

Fact 6.9 (Clause with New Predicate Is Conservatively Extending)
In monotonic classical logic:

Any clause that contains at least one literal in a new predicate symbol is a conservatively extending update.

I.e., in Definition 3.6, let $U[P, Q]$ be

$$\forall x.\ \sigma_1 P1(t_1) \vee \ldots \vee \sigma_m Pm(t_m) \vee \sigma_{m+1} Q1(t_{m+1}) \vee \ldots \sigma_{m+k} Qk(t_{m+k})$$

where k is at least 1. Here, for each $i = 1, \ldots, m$: $Pi \in P$. For each $j = 1, \ldots, k$: $Qi \in Q$. For each $h = 1, \ldots, m+k$: σ_h is a sign (either positive or negative), and t_h is a term in the tuple of individual variables x.

Then $U[P, Q]$ is a conservatively extending update.

This fact is related to the fact that explicit definitions of new symbols are always conservatively extending (Fact 3.7). A clause (e.g., rule) mentioning a new symbol can be viewed as a partial definition; an explicit definition can often be viewed as a conjunction of necessary and sufficient rules.

Proof : Choose any of the Q-literals. For any P, one can always choose Q to be such that this literal is true for any instantiation of x. □

Example 6.10 (Animals, Conservatively Extending Updates)

Continuing Example 6.5:
Next, suppose there is another update $\mathcal{U}_{10}$:

$\bullet\!\!>\ \forall x.\, cat(x) \supset mammal(x)$

This new base rule has a formula that is a clause containing exactly one appearance of a new literal: *cat*. By Fact 6.9, the update's base formula thus conservatively extends the previous base. Therefore, Theorem 6.8 tells us that this update is monotonic relative to the Animals slice. By our disjoint-sub-languages results (Corollary 6.3), since the update affects only the Animals slice, all of the other slices are left safe. Thus the update is guaranteed by our results to be globally monotonic.

This once again illustrates the **synergy** created by combining **four different kinds of special-case decompositions**, to get more locality, e.g., clean decomposition, than is available from any one alone. Here, we are using three different special cases (conservative extension for Animals, total propositional for Feelings, and total prioritization over modules for Meetings) to decompose more finely, i.e., hierarchically, within another special case (disjoint sub-languages).

Next, suppose there is another update $\mathcal{U}_{11}$, containing the two defaults that bats squeak when awake, and that bats are nocturnal:

$(a4) :>\ bat(x) \wedge awake(x, time) \supset squeaks(x, time)$

$(a5) :>\ bat(x) \supset nocturnal(x, time)$

$\bullet\!\!>\ \forall x.\, nocturnal(x) \equiv (awake(x,t) \supset dark(x,t))$

In this case, Theorem 6.8 has nothing directly to say, since it only applies to updates consisting of a single base or default axiom. However, a **useful analytic technique is to serialize an update consisting of several axioms** into smaller-grain updates.

In this example, we can choose to view the update as three single-axiom updates. Using the Facts about conservative extension (explicit definitions and clauses with one new literal), it follows that each of these single-axiom updates *is* conservatively extending cf. Theorem 6.8, hence guaranteed to be monotonic relative to the Animals slice. There is a bit of a subtlety, though: the base axiom update must come last in the sequence of the single-axiom update in order to apply Fact 3.7. Therefore, the overall three-default update ($\mathcal{U}_{11}$) is: monotonic relative to the Animals slice; and, by our disjoint sub-languages results, globally monotonic.

Serializing: An Analytic Technique:
More generally, one can try to analyze the effect of any multi-axiom update by serializing it: i.e., by choosing a partition of the axioms in that update, and some sequence over that partition.

6.5 Sympathetically Solitary

In this section, we find a non-conflicting case of sympathy that generalizes Lifschitz' "solitary" class. In this case, we are able to give a first-order "explicit solution" to the entire circumscription. This solution is computable in polynomial time. This implies the opportunity to develop tractable and decidable inference and belief revision algorithms for further specializations of this "sympathetically solitary" class.

The explicit solution also implies strong (global) safeties of updating.

The non-conflict in the sympathetically solitary case is at finest-grain: the explicit solution corresponds to a **finest-grain clean** conjunctive decomposition with one default (or default instance, essentially) per slice.

Sympathetically solitary is a restriction on the entire axiom set. Perhaps the most important aspect of this restriction is that **default rules cannot be chained: the default reasoning is "shallow"**.

Definition 6.11 (Solitary (Lifschitz))

Lifschitz [1985] defined a predicate circumscription to be *solitary* when the base sentence has the form:

$$B[P,W] \;\equiv\; (L[W] \leq P) \;\wedge\; NEG[P,W]$$

where P are the minimized predicates, W are the fixed predicates, and the predicates P appear only negatively in NEG. Note that it is important that L mentions only fixed symbols. (As usual, P and W are presumed to be distinct. "L" is mnemonic for "lower" bound on P.) He did not permit **auxiliary** (recall page 88 for the terminology "auxiliary") predicate variables, and considered only the layered case of priority.

Terminology: Positivity / Negativity: By positive / negative appearance, we mean the standard logical notion. We say that the predicate symbols Q appear *positively* (respectively, *negatively*) in the formula $E[Q,Y]$ when E is equivalent to a formula in which the only logical connectives are $\neg$, $\wedge$, and $\vee$, and every appearance of each member of Q appears in the scope of an even (respectively, odd) number of negations.

Theorem 6.12 (Solitary (Lifschitz))

Lifschitz [1985] showed that, in this solitary case:

$$PPC(B;P;R;fix\,W;P,W) \;\equiv\; B[P,W] \wedge (P = L[W])$$

(where, remember, there are no auxiliary variable predicates, and R is layered). Equivalently, the $=$ above can be replaced by $\leq$. Note that the prioritization R is irrelevant: essentially, because there is no conflict (see below).

As Lifschitz noted (and as Reiter did, for a special case, before him), this case of circumscription thus bears a close relationship to the *predicate completion* of [Clark, 1978]. Lifschitz' result generalizes a previous result by Reiter [1982] for the case of a single minimized predicate.

Observation 6.13 (Nice Features of the Solitary Case)

The solitary case displays several desirable, interesting features:

1. **Conflict-Free:** There is no conflict between defaults (i.e., between minimizations of different predicates). Intuitively, "there is a unique minimal model", corresponding, intuitively, to a single extension in the sense of Default Logic or Autoepistemic Logic. (More precisely: for any given partial model of the fixed symbols (including choice of domain), there is a unique minimal model that extends it. E.g., if there are no fixed predicates, and there is domain closure plus a finite Herbrand base plus uniqueness of names, then there is a unique minimal Herbrand model.) In this regard, the solitary case is similar to the stratified case of logic programs with negation (which is discussed in section 8.7). This indicates that the solitary case is a prospective "easy" case of NMR capable of relatively efficient implementation and practical usage, in the way that stratified logic programs with negation are. (See section 8.7 for discussion of logic programs and their relationship to PPC.)

2. **Explicit Solution and Collapse to First-Order:** Theorem 6.12 gives a relatively **explicit solution** (**terminology**) to the circumscription. If B is first-order, then so is the circumscription, by the right-hand side of the theorem. In the circumscription literature, one says that the circumscription **collapses** (**terminology**) to first-order. I.e., the non-monotonic part of the circumscriptive theory (corresponding to the circumscription's augmentation part) can be axiomatized in first-order: $P \leq L[W]$. Recall (sub-section 4.3.1) that in the circumscription literature, one says that this is the **generating axiom** for the non-monotonic conclusions. (Recall that in "generating axiom", "axiom" is meant in the monotonic-logical sense, *not* in the sense of being part of a NM axiom set.) If L is quantifier-free, then the $L[W] \leq P$ part of the base corresponds to universal rules (or clauses). In that case, then the generating axiom $P \leq L[W]$ also corresponds to universal rules (or clauses). Thus the circumscriptive theory has a relatively simple, explicitly-determinable form (of generating axiomatization) in this case. (Also, note that the collapse to first-order has implications for completeness of inference, due to the incompleteness of second-order logic with standard models.)

3. **Fine-Grain Clean Decomposition:** One way to view Theorem 6.12 is that the global circumscription can be cleanly conjunctively decomposed into one slice per default (i.e., per minimized predicate):

$$PPC(B; P; R; fix\ W; P, W) \ \equiv \ \bigwedge_{i \in N} PPC(B; Pi; \emptyset; fix\ W; P, W)$$

Indeed, the global is, essentially, cleanly slice-able with even finer grain: by default instance (Definition 2.46), as we show in Corollary 6.20.

4. **Easy To Recognize Syntactically:** The solitary case can be recognized syntactically relatively easily in many cases of interest (e.g., clausal base and defaults), : with computational time complexity that is, worst-case, $O(n \cdot log(n))$ in the size of the global CLD axiom set, as we show in sub-section 7.7.11.

We will aim to make the most of these advantages. In order to do so, though, we will generalize the solitary case in several ways, leading, finally, to the *sympathetically solitary* case.

Limitation of Previous Solitary Case: Auxiliaries:
As Lifschitz realized, the solitary case's lack of auxiliary predicate variables is a major handicap to its applicability. He pointed out that this obstacle can often be got around by first using another result about eliminating auxiliary variables.

Lemma 6.14 (Eliminating Auxiliaries (Lifschitz))
Lifschitz showed ([1985] Proposition 2') that

$$PPC(B[P,Y,W];P;R;fix\ W;P,Y,W) \quad \equiv$$
$$B[P,Y,W] \ \wedge \ PPC((\exists Y'.\ B[P,Y',W]);P;R;fix\ W;P,W)$$

for layered R. We observe that this generalizes straightforwardly to *non*-layered priority as well.

This method performs existential projection of the *entire* base onto the auxiliary-free sublanguage (recall the discussion about existential projection in connection with conservative extensions cf. Definition 3.6). While feasible by hand for small examples, this existential projection appears, to us, computationally impractical for most large base axiom sets. Therefore, we would like to find some other, more practical method of exploiting the nice features of the solitary case.

Definition 6.15 (Variable-Solitary)
The first step of our generalization of the solitary case is to permit auxiliary predicate variables directly in its form; i.e., in Definition 6.11 to permit additional (distinct) predicate variables Y to appear in NEG. (Lifschitz did not, and no one else did either, previously.) The second step of our generalization is to permit non-layered prioritization. After these two steps, we define the more general case as: *variable-solitary*

Observation 6.16 (Limited Applicability of (Even Variable-) Solitary)
Unfortunately, the applicability of the (variable-)solitary case is still limited, even with auxiliaries (and non-layered priority). The most common application of predicate circumscription, especially for default reasoning, e.g., default inheritance, is abnormality theories (Definition 3.5). Yet they are not covered: whether in adaptive or definitional form, the abnormality predicates appear positively in the base in mixture with the auxiliary predicates.

Stratified logic programs with negation are another important application of prioritized predicate circumscription (see section 8.7). Like the solitary case, they are conflict-free. However, in stratified logic programs with negation, there are (typically) no fixed predicates. (And no implicit fixtures are possible, due to lack of conflict.) Thus the only time they are solitary is when all clauses have empty tails. This is, essentially, just equivalent to a simple case of the Closed World Assumption. (Remember, all predicates are being minimized.) In short, it gives no insight there.

Next, we generalize the idea of (variable-)solitary by modifying it in a more radical fashion than just permitting auxiliaries, so as to arrive at a similarly nice (cf. Observation 6.13) special case that is relatively easy to detect syntactically (indeed, essentially as easy as for the original solitary case), yet applies to prioritized default and abnormality theories. We call this case: sympathetically solitary.

The basic idea is to impose a solitary-like condition on each of the predicates that appear in the default formulas. For each predicate, the direction of this condition corresponds to the sign with which that predicate appears in the defaults, i.e., it is "sympathetic" towards the defaults' maximization. We require, moreover, that this sign be unique. Note that variable-solitary is the special case of sympathetic-solitary in which the default formulas are negated predicates.

Definition 6.17 (Sympathetically Solitary)
In a prioritized default circumscription $PDC(B; D; R; fix\ W; Z)$ where the only fixing is of symbols, let Q stand for the tuple of varied (i.e., non-fixed) predicates that appear in the default formulas D. Suppose that, for each predicate $Qk \in Q$, every appearance of Qk in D has the *same* sign (alias, polarity). Let σ_k stand for this sign. Let σ stand for the tuple of the σ_k's.

Suppose also that the base has the form:

$$B[Q, Y, W] \ \equiv\ (\sigma Q \leq G[W]) \ \wedge\ S[Q, Y, W]$$

and S is positive in σQ. Here $Y \overset{\text{def}}{=} Z - Q - W$ are the varied predicates that do not appear in D. By S being positive in σQ, we mean that for each Qk, S is positive in Qk if σ_k is positive, and S is negative in Qk if σ_k is negative. I.e., every appearance of Qk in S has the same sign, and that sign is σ_k (the same sign as Qk appears with in D).

Then we say that the PDC is in *sympathetically solitary*, or *sympathetic-solitary*, form.

Example: See Example 6.27.

Theorem 6.18 (Explicit Solution, Sympathetic-Solitary)
Suppose that a prioritized default circumscription is sympathetically solitary cf. Definition 6.17. Then

$$PDC(B; D; R; fix\ W; Q, Y, W) \ \equiv\ B[Q, Y, W] \ \wedge\ (D[Q, W] = E[W])$$

where E is the result of substituting $G[W]$ for σQ in D. (Equivalently, the $=$ above can be replaced by a $\geq$.)

Note that, as with the solitary and variable-solitary cases, the prioritization R is irrelevant.

Proof Overview: Uses analytic techniques and lemmas (about the properties of prioritized pre-orders) that we developed to prove our results about general-case canonical decomposition along prioritization. The argument involves substitution and properties of positivity / negativity. See Appendix. □

Intuition: Predicates Q Pull Together Towards Bounds:
Theorem 6.18 tells one that, in the sympathetically solitary case, the variable predicates Q mentioned in the defaults all pull (i.e., rise modulo their appearing sign σ) towards their (upper) bounds

G. Intuitively, each appearing predicate "helps out in behalf of the defaults" as much as it can, given the constraints of the base and fixtures. The non-monotonic conclusions of the circumscription then are exactly the "direct" effect of those predicates' "efforts" on the default formulas (i.e., via the substitution in the statement of Theorem 6.18). The requirement that each predicate in Q appear in the defaults with a unique sign ensures that no predicate in Q is "confused" about which direction to pull. The requirement that G is fixed ensures that these bounds themselves are not non-monotonically affected in the process, i.e., that they are "hard" bounds. The requirement that Q not appear in S with sign opposite to σ helps to ensure that there is no conflict in the pulling process, i.e., that the different predicates in Q are sympathetic towards each other.

Note that we can apply these same intuitions towards the solitary and variable-solitary cases.

One can view the (Q-) condition on the base in sympathetic-solitary as, essentially, the same as in variable-solitary, modulo flexibility about the signs of Q.

Observation 6.19 (Expressive Analysis, Sympathetically Solitary)
"Wow, sounds great; maybe TOO good; what can I really use it for?" The explicit solution above is our central result about sympathetic-solitary. Below we give some of its ramifications, and some examples. But, first, we pause to take stock of what kind of reasoning can be expressed in the sympathetically solitary form / class. Utter lack of conflict between defaults should make us suspect that the class is highly restrictive.

One piece of good news, especially compared to the (variable)solitary class, is that the sympathetic-solitary class can express interesting default formulas, e.g., corresponding to default inheritance.

However, as we mentioned earlier, perhaps the most significant restriction of sympathetic-solitary is that defaults cannot be chained in the sense of backward or forward chaining of rules in inference.

More precisely, let all the base and default formulas be clausal, and let the only fixtures be of functions. Each clause can be viewed as a rule. Then no default rule can participate in an inferential chain, even if the other rules in such a chain might be base rules rather than default rules. The sympathetic-solitary restriction is actually even a bit stronger: there can be no chaining through any predicate that appears in a default rule. I.e., no predicate that appears in a default rule can have a literal resolved upon by the resolution inference rule in first-order or propositional (base-language) logic.

There is some more good news, though.

With more fixtures, the picture is a bit brighter: chaining can take place through any literal whose predicate is fixed, even if that predicate or literal appears in a default rule.

Compared to default inheritance, e.g., cf. [Touretzky, 1986][3], the sympathetic-solitary class is more expressive in several respects. A default or base rule may have several antecedents; i.e., as a clause it need not be binary. And the rule need not be Horn: negation may appear freely. And it

[3]Recall also the footnote on page 15.

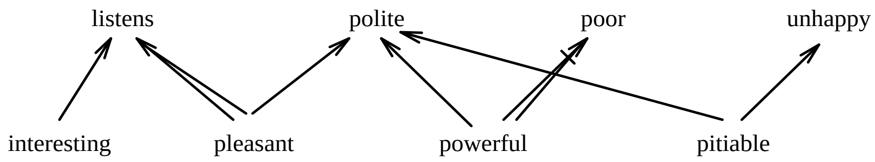


Figure 6.1: Shallow Topology of Sympathetically Solitary Class.

may involve relational (arity > 1), not just monadic (1-ary), predicates. Indeed, the defaults and base formulas in sympathetic-solitary need not be clausal at all: they may be arbitrary first-order formulas. This is more expressive than Prolog-style logic programs too.

Compared to Horn: even in the clausal case, sympathetic-solitary is plain different. Horn form does permit chaining. But it does not permit disjunctions of several literals with positive sign. By contrast, sympathetic-solitary does permit such disjunctions; indeed, it permits clauses that contain several positive literals in the same predicate.

Thus the major issue in sympathetic-solitary is what we might call the "topology" of interaction between the default (and base) formulas. In the case of binary clauses with no fixed predicates, we can illustrate matters graphically, in the manner usual for inheritance networks and semantic networks. Figure 6.1 illustrates, in this manner, the shallow "look" of inheritance nets that are permitted by the sympathetic-solitary property. It illustrates a simplified version of Example 6.27 later in this section. Illustrated, in conventional inheritance net / semantic net style, is a simplified version of the base and default rules from the sympathetically solitary Example 6.27. Nodes are labelled with predicate names, and indicate literals. An arrow indicates a rule, i.e., an implication. Single lines stand for default rules; double lines stand for base rules. A cross-hatch stands for implication of a negation. □

The explicit solution implies a clean, finest-grain conjunctive decomposition: (since) there is no conflict between any two defaults, nor, indeed, between any two default instances. That is, there is **no conflict at all** between defaults.

Corollary 6.20 (Clean Finest-Grain Conjunctive Decomposition)

Suppose that a prioritized default circumscription is sympathetically solitary cf. Definition 6.17. Then it has a clean finest-grain conjunctive decomposition, as follows.

One by One: Let K be the index tuple of Q. Let J be the index tuple of W.
In slice i there is exactly one default: Di (and no prioritization).
Let $Ki \subseteq K$ index the members of Q^K that actually appear in the formula Di. Let $Ji \subseteq J$ index the members of W^J that actually appear in either the formula Di or the formulas G^{Ki}. The base for slice i is

$$BGi[Z] \stackrel{\text{def}}{\equiv} (\sigma Q)^{Ki} \leq G^{Ki}[W^{Ji}]$$

The fixtures for slice i are: W^{Ji}.

I.e., the PDC for slice i is:

$$PDC(BGi; Di; \emptyset; fix\, W^{Ji}; Z)$$

In addition, there is one more slice, containing only the base axiom S.

The conjunctive decomposition equivalence is:

$$PDC(B; D; R; \mathit{fix}\ W; Z) \ \equiv \ S[Z] \ \wedge \bigwedge_{i \in N} PDC(BGi; Di; \emptyset; \mathit{fix}\ W^{Ji}; Z)$$

where N is the index tuple of D.

More generally, any of the rest of the global base and fixtures may also be included in each slice.

Instance Grain: Suppose furthermore that the base B entails domain closure. By Theorem 2.47, each open default Di in D is thus equivalent to a module: its default instances in parallel.

Then, in the above one-by-one decomposition, the slice for each open default can be cleanly conjunctively decomposed, further, into one slice per instance.

In (somewhat tedious) detail: Let ah stand for the h^{th} tuple, of domain elements, of the appropriate arity for the open default Di. That is, let $Di[Q^{Ki}, W^{Ji}, ah]$ be a default instance (default formula). Let us write this as Dih for short.

In the h^{th} instance-grain slice for the open default Di, there is exactly one default: Dih.

Let us call this slice ih.

The base for slice ih is:

$$BGih[Z] \ \stackrel{\text{def}}{\equiv} \ (\sigma Q)^{Ki}(ah) \ \leq \ G^{Ki}[W^{Ji}, ah]$$

The fixtures for slice i are: W^{Ji}.

I.e., the PDC for slice ih is:

$$PDC(BGih; Dih; \emptyset; \mathit{fix}\ W^{Ji}; Z)$$

The conjunctive decomposition equivalence is:

$$PDC(BGi; Di; R; \mathit{fix}\ W^{Ji}; Z) \ \equiv \bigwedge_{h \in M} PDC(BGi; Di; \emptyset; \mathit{fix}\ W^{Ji}; Z)$$

where M indexes the instances.

Note that this instance grain result depends, because Theorem 2.47 does, on the **assumption** (page 36) that all functions are fixed relative (Definition 3.47) to the circumscription. (But recall that, by, Theorem 2.24, uniqueness of names or a complete theory of equality implies the fixture of all functions.)

Proof : Apply Theorem 6.18 to each slice. The decomposition equivalence then follows. For the further instance grain decomposition, also use Theorem 2.47. □

Intermediate-Grain Too:

Yet more generally, one may choose any set of subsets of the defaults above, whose union includes all of the global defaults, as a basis for intermediate grain size clean conjunctive decomposition.

Theorem 6.18 can be applied to abnormality-style prioritized *predicate* circumscriptions.

Corollary 6.21 (Explicit Solution, Abnormalities, Sympathetic-Solitary)

Suppose that, in an abnormality-style prioritized predicate circumscription $PPC(B; ab; R; fix\ W; ab, Q, Y, W)$, the base has the form

$$B[ab, Q, Y, W] \ \equiv \ (ab \geq \neg D[Q, W]) \ \wedge \ (\sigma Q \leq G[W]) \ \wedge \ S[Q, Y, W]$$

or the form

$$B[ab, Q, Y, W] \ \equiv \ (ab = \neg D[Q, W]) \ \wedge \ (\sigma Q \leq G[W]) \ \wedge \ S[Q, Y, W]$$

where D, σ, Q, G, and S are as in Theorem 6.18. Then

$$PPC(B; ab; R; fix\ W; ab, Q, Y, W) \ \equiv$$
$$B[ab, Q, Y, W] \wedge (ab = \neg D[Q, W]) \wedge (D[Q, W] = E[W])$$

where E is as in Theorem 6.18.

Proof : Immediate from Theorem 6.18 (default case) plus Corollary 3.46 (conservative extension for abnormalities). □

Defaults Versus Minimized Predicates:
Note that solitary (and variable-solitary) is *not* a special case of this abnormality version of sympathetically solitary. Thus Corollary 6.21 is an interesting generalization for the predicate case. This also shows the advantage of thinking in terms of prioritized *default* circumscriptions, not just prioritized predicate circumscriptions.

Corollary 6.22 (Monotonicity of Sympathetic Base Update)
In Theorem 6.18 (explicit solution) and Corollary 6.21 (abnormality-form), updating the base with $U[Q, Y, W]$ is (globally) monotonic if U is positive in σQ (see property of S in Definition 6.17).
Terminology: In this case we say that the update is **sympathetic** to the previous axiom set / theory / defaults.

Proof : Immediate from Theorem 6.18: consider U as part of S. □

Interestingly, and less straightforwardly, the same sympathy condition on the base update also ensures monotonicity for a default update with any prioritization. This accords with our intuition that base updates can be viewed as a special case of prioritized default updates (recall Observation 5.74 and Theorem 5.75).

Theorem 6.23 (Monotonicity of Sympathetic Default Update)
In Theorem 6.18, updating with a new default whose formula part is $U[Q, Y, W]$, with any prioritization for that new default relative to the previous defaults, is globally monotonic if U is sympathetic, i.e., positive in σQ (see property of S in Definition 6.17).

This result applies to abnormality theories as well, cf. Corollary 6.21.

Proof Overview: In addition to Theorem 6.18, we use canonical conjunctive decomposition for modules: Corollary 5.28. We first show the case where the new default is added in parallel to all the previous defaults, then use the monotonicity of prioritization: Theorem 2.58. See Appendix. □

Defaults Versus Minimized Predicates:
The above result also shows the advantage of thinking about updates in terms of prioritized *default*

circumscriptions and CLD, not just prioritized predicate circumscriptions: the similarity of default updates to base updates is manifested more clearly.

Observation 6.24 (Redundancy of Prioritization Update)
In the sympathetically solitary case (default or abnormality form):
Since prioritization is irrelevant, it is redundant as an update. This redundancy is forever for updates that maintain the sympathetically solitary property.

Proof : Immediate from Theorem 6.18. □

Another nice property is that: Typically in applications we have looked at, a base update that preserves the sympathetic-solitary condition only affects some of the finest-grain slices. The scope of this influence can, moreover, often be detected syntactically relatively easily. The conclusions in the rest of the slices are safe under the update. The following corollary makes this safety precise; our example a bit later on illustrates how detection can sometimes be easy.

Corollary 6.25 (Selective Safety, Base Updates)
In Corollary 6.20:
Suppose that a base update preserves the sympathetic-solitary condition. Suppose also that it only affects some of the bounding expressions: i.e., suppose the update changes only G^U for some $U \subset K$.
Let S index the subset of the defaults whose default formulas mention no symbols from Q^U. Suppose, furthermore, that S is not empty.
Then all conclusions in the conjunctive combination of the finest-grain slices corresponding to S are safe under the update.

Suppose also that the previous global base entails domain closure and uniqueness of names. (Note that by Theorem 2.24, this implies that all functions are fixed(-relative).) The instance-grain decomposition in Corollary 6.20 thus holds. Suppose, furthermore, that the base update is ground, and mentions only a subset V of the domain elements.

Then all previous ground conclusions which do not mention those domain elements V are safe under the update.

Proof : Apply differential decomposition (Observation 4.6) to Corollary 6.20. □

Observation 6.26 (Nice Features of Sympathetically Solitary)
The sympathetically solitary case of prioritized *default* circumscription has **all the nice features of the solitary case** that we discussed in Observation 6.13. Plus it is more applicable, especially to default reasoning. It thus makes an interesting restricted target class / form for specification.

Next, we mention some **more details**, organized according to Observation 6.13:

2. **Rule Form and Explicit Solution:** In Theorem 6.18, note that $E[W]$ is first-order (and universal) if $D[Q, W]$ and $G[W]$ are. Thus if the base and defaults in the PDC are first-order (universal) then the PDC **collapses to first-order (universal)**. If the bounding expressions

G are quantifier-free, then the $\sigma Q \leq G[W]$ part of the base corresponds to universal rules (or clauses). (We can write this, suggestively, in contrapositive form as: $GG[W] \leq \neg\sigma Q$, i.e., as implications by fixed expressions of "violations" of σQ. Here $GG[W] \stackrel{\mathrm{def}}{=} \neg G[W]$.) In that case, then the generating axiom $\sigma Q \geq G[W]$ (contrapositively: $GG[W] \geq \neg\sigma Q$) also corresponds to universal rules (or clauses).

4. **Easy to Recognize Syntactically:** Since the sign-of-appearance conditions are about predicate symbols, not formulas, sympathetic-solitary form is, essentially, no harder to detect than (variable-)solitary form. The explicit solution in Theorem 6.18 can be computed in time that is, worst-case, **polynomial** in the size of the CLD axiom set. In sub-section 7.7.11, we discuss algorithms and computational complexity.

Example 6.27 (Socializing, Sympathetically Solitary)

Continuing Example 6.10:

Next, suppose there is another update $\mathcal{U}_{12}$, about Socializing. This update contains two defaults: that pleasant-ness is returned by politeness, and that powerful people are treated politely. And it contains some assorted for-sure information, including information about the predicates mentioned in those defaults.

$(s1) :> \ pleasant(x, y) \supset polite(y, x)$

$(s2) :> \ powerful(x) \supset polite(y, x)$

$\bullet\!> \ +pleasant: \ \ \langle Larry, Jack\rangle, \langle Peg, Ed\rangle, \langle Ramon, Killjoy\rangle,$
$\langle Srid, Alberto\rangle, \langle Alberto, Srid\rangle,$
$\langle Srid, Wlod\rangle, \langle Wlod, Srid\rangle$

$\bullet\!> \ +powerful: \ \ Bush, Gore, Cuomo, Gotti, Bonnie, Clyde$

$\bullet\!> \ \neg polite: \ \ \langle Jack, Larry\rangle, \langle Ed, Peg\rangle, \langle Killjoy, Ramon\rangle, \langle Livia, Claudius\rangle$

$\bullet\!> \ +polite: \ \ \langle Peg, Maggie\rangle, \langle Maggie, Peg\rangle, \langle Ollie, Kevin\rangle$

Then Socializing is partitionable by disjoint sub-languages from the previous global axioms. (Though some of the function symbols, e.g., Ed, also appear in the previous axioms, that is OK, since the functions are fixed in this example.) Thus the update is globally monotonic, and the Socializing group of axioms forms a clean slice. This behavior is similar to several previous updates in our running example.

What's new about the behavior of *this* update in our running example is:
the Socializing slice is sympathetically solitary. Thus, by Theorem 6.18, there is no conflict at all between the defaults in the Socializing slice, and there is an explicit solution to the Socializing slice circumscription. Corollary 6.20 says that this explicit solution is itself conjunctively decomposable with finest grain. (Instance grain is available in this example since there is domain closure and uniqueness of names.) Consider an intermediate grain size: one slice per open default. The slice for the pleasant-ness default $(s1)$, within the Socializing slice, yields the **generating axiom** (recall sub-section 4.3.1):

$\forall x, y.\ \neg\{[(x{=}Larry) \wedge (y{=}Jack)] \vee [(x{=}Peg) \wedge (y{=}Ed)]\}$

$\wedge\ pleasant(x, y)$

$\supset\ polite(y, x)$

I.e., the default "goes through" for every instance except for the two for-sure violations.

The slice for the powerful-ness default $(s2)$, within the Socializing slice, yields the **generating axiom** (recall sub-section 4.3.1):

$\forall x, y.\, powerful(x) \supset polite(y, x)$

For this default rule, there are no for-sure exceptions/violations.

Together with the Socializing base axioms, the above two conclusions monotonic-logically entail (i.e., generate) *all* of the conclusions in the Socializing slice, i.e., the entire Socializing slice theory.

Next, suppose there is another update $\mathcal{U}_{13}$ about Socializing;

$(s3) :>\ pitiable(x) \supset polite(y, x)$

$(s4) :>\ pitiable(x) \supset (unhappy(x) \vee \neg healthy(x))$

$(s5) :>\ interesting(x) \wedge \neg hurried(y)\ \supset\ listens(y, x)$

$\bullet\!>\ \forall x, y.\, obsequious(x, y) \supset polite(x, y)$

$\bullet\!>\ \neg pitiable(Fred) \vee \neg powerful(Fred))$

$\bullet\!>\ \forall x.\, powerful(x) \supset (big(x) \vee bad(x))$

$\bullet\!>\ \forall x, y.\, pleasant(x, y) \supset listens(x, y)$

$\bullet\!>\ \forall x.\, \neg violent(x) \wedge young(x) \wedge poor(x)\ \supset\ \neg powerful(x)$

$\bullet\!>\ \forall x.\, scared(x) \supset \neg violent(x)$

$\bullet\!>\ +young:\ \ Ellie, Fred, Ginger$

$\bullet\!>\ +scared:\ \ Dara, Ellie, George$

$\bullet\!>\ +poor:\ \ Bender, Ellie$

$\bullet\!>\ +obsequious:\ \ Harry, Io$

$\bullet\!>\ interesting(Houk)$

$\bullet\!>\ \neg hurried(Hum)$

$\bullet\!>\ hurried(Rafaella) \vee hurried(Randi)$

This update is **sympathetic**. More precisely, each of the base and defaults in it is sympathetic. *Serializing* $\mathcal{U}_{13}$ (see Example 6.10) enables the application of Corollary 6.22 (for the base axioms in the update) and Theorem 6.23 (for the default axioms in the update). They imply that the update $\mathcal{U}_{13}$ is monotonic with respect to the Socializing slice. Together with the overall disjoint-sub-languages clean slicing, this implies that the update is globally monotonic.

Next, suppose there is another update $\mathcal{U}_{14}$ about Socializing;

$\bullet\!>\ \neg polite(Killjoy, Ramon)$

This update is non-monotonic: it causes the retraction of the previous default conclusion $polite(Killjoy, Ramon)$.

However, this update preserves the sympathetically solitary property of the Socializing slice. And it is ground: mentioning only the two individuals Killjoy and Ramon. Moreover, instance-grain clean conjunctive decomposition is available, by our sympathetic-solitary results, for this example. Thus Corollary 6.25 (selective safety, base updates) implies the safety of all previous conclusions in the Socializing slice that do not mention Killjoy or Ramon. Together with the overall disjoint-sub-languages decomposition, this implies almost all of the previous conclusions are guaranteed safe for this update, by our results.

This once again illustrates the **synergy** created by combining five different kinds of special-case decompositions, to get more locality, e.g., clean decomposition, than is available from any one alone. Here, we are using four different special cases (sympathetic solitary for Socializing, conservative extension for Animals, total propositional for Feelings, and total prioritization over modules for Meetings) to decompose more finely, i.e., hierarchically, within another special case (disjoint sub-languages).

Example 6.28 (Socializing, NOT Sympathetically Solitary)
Continuing Example 6.10:
Next, suppose there is another update $\mathcal{U}_{15}$, about Socializing, containing the following axioms. Each of these axioms has the property that, even considered as a single-axiom update, it destroys the sympathetically solitary property of the Socializing slice.

$(s6) :> \; polite(x, y) \supset good\_tempered(x))$

$(s7) :> \; listens(x, y) \supset understands(x, y)$

$\bullet\!> \; \forall x.\, homeless(x) \supset pitiable(x)$

$\bullet\!> \; pleasant(Karen, Srid) \vee pleasant(Gus, Srid)$

6.6 Strong Sympathy

Strong Sympathy:
In this section, we show that another non-conflicting case of sympathy is when new base or default formulas are formed purely and positively from previous default and fixed formulas. Such "strongly sympathetic" updates are globally and forever (Definition 4.17) monotonic.

Intuition: Confirming:
The intuition behind strong sympathy is that a strongly sympathetic update is *confirming* in a particular sense. In the case of a base update, it is confirming what the previous defaults have already been "trying to say" or "pushing for". In the case of a default update, it is confirming in a weak sense: echoing. The sympathy is *strong* (and strongly restrictive) in the sense that the new belief can be put purely "in terms of" the previous premise beliefs.

Theorem 6.29 (Monotonicity of Strongly Sympathetic Updates)
Let $PPC(B; P; \emptyset; fix\ W; P, Y, W)$ be a parallel predicate circumscription. Suppose that the sentence $U[P, W]$ is negative in the minimized predicates P, and mentions only P and the fixed symbols W (no auxiliaries may appear). Then updating the base with U is globally monotonic.

More generally, let $PDC(B; D; \emptyset; fix\ F; Z)$ be a parallel default circumscription. Suppose that the formula $U[Z]$ is tautologically equivalent to a formula that is formed purely and positively from the default formulas D plus fixed formulas. By "fixed formulas" here, we mean formulas that are fixed relative (Definition 3.47) to the circumscription: e.g., formulas formed from the explicitly fixed formulas F. (See also Definition 3.50.)

We **define** such a U to be **strongly sympathetic** to the circumscription, and to the defaults.

Then:

1) If U is closed, then updating the **base** with U is globally **monotonic**. In this case, the circumscription after the update is just U conjoined with the previous circumscription.

$$PDC(B \wedge U; D; \emptyset; fix\ F; Z) \equiv U[Z] \wedge PDC(B; D; \emptyset; fix\ F; Z)$$

And, interestingly:

2) Updating with U as a new **default** is **redundant** (Definition 4.17):

$$PDC(B; D, U; \emptyset; fix\ F; Z) \equiv PDC(B; D; \emptyset; fix\ F; Z)$$

Note that in the case of a default update, U may be an open formula.

Moreover, in terms of the special case of CLD without priorities, this monotonicity and redundancy are **forever** (Definition 4.17).

Note that the above depends on the **assumption** (page 36) that all functions are fixed. More generally, if only some functions are fixed, then, in the forming, only functions that are indeed fixed may appear. That is, the instantiation aspect of forming must be limited to employ only the subset of the functions that are indeed fixed.

For the case of a **base** update: the above can be viewed as a **clean conjunctive decomposition** in which there are two slices. The entire previous axiom set is the first slice. The update is the second slice. Likewise, the above can be viewed as a **clean serial decomposition**, with the same constituent axiom sets, in either sequence.

(For the case of a **default** update, the above can be viewed as a **clean serial decomposition**, with the update coming seqentially after the entire previous axiom set. This holds for any monotonic update.)

The non-conflict in this conservative extension case has mixed grain: it is between one (the update) and a group (the previous axioms). Note that strong sympathy is a restriction on the update, relative to (i.e., applicable to) *any* previous axiom set that is *parallel*.

Proof Overview: We use a generalization of positivity / negativity, and a lemma relating that to forming. In the default case, we show that the maximized global pre-order after the update is equivalent to that before the update. See Appendix. □

Corollary 6.30 ("And" & "Or" "Rules" In Updating)

Theorem 6.29 implies that updating with the conjunction, or the disjunction, of partially instantiated previous defaults is redundant as a default update, and monotonic as a base update. E.g., in a parallel default theory, let U be $D1 \wedge D2$ or $D1 \vee D2$, where $D1$ and $D2$ are previous default formulas. Then U is redundant as a default update, and monotonic as a base update.

The above depends on the **assumption** that all functions are fixed, and can be generalized when that assumption is relaxed, in the same way that Theorem 6.29 does.

Proof : Immediate as special case of Theorem 6.29. □

Intriguingly, one can view this in terms of a **natural-deduction-style system for inferring redundancy or monotonicity**. E.g.:

$:>\ D1$

$:>\ D2$

$- - - - - - - -$

$:>\ D1 \wedge D2$

and:

$:>\ D1$

$:>\ D2$

$- - - - - - - - - - - - - - - - -$

$monotonicity\ of\ \ \bullet>\ D1 \wedge D2$

Note that many of our other results, e.g., about forming fixed formulas, can be also viewed as corresponding to natural-deduction-style inference rules.

Previous Results:
Theorem 6.29 generalizes a previous result by Lifschitz [1990a] which covered, in effect, base updates in the predicate case without auxiliary predicate variables.

Defaults Versus Minimized Predicates:
Theorem 6.29 and Corollary 6.30 again show the advantage of thinking about updates in terms of prioritized *default* circumscriptions and CLD, not just prioritized predicate circumscriptions. In Theorem 6.29, the similarity of default updates to base updates is manifested more clearly. In Corollary 6.30, the abnormality-theory version is much more complicated to state and less illuminating.

Generalization to Prioritized Case:
Elsewhere (work in preparation), we will show how Theorem 6.29 generalizes to the prioritized case. Matters become complicated by the need to modify positivity / negativity to take into account the prioritization. We do not take the space here for details; it would be too much of a diversion from our main focus in this thesis.

Example 6.31 (Socializing, Strong Sympathy)

Continuing Example 6.28:
Next, suppose there is another update $\mathcal{U}_{16}$ about Socializing:

$\bullet>\ \forall y.\, polite(Fred, y) \supset good\_tempered(Fred)$

$(s8) :>\ listens(Ed, Peg) \supset understands(Ed, Peg)$

Here, the base axiom's formula is formed by simple instantiation from the previous default's $(s6)$ default formula. Likewise, the update default axiom's formula is formed by simple instantiation from another previous default's $(s7)$ default formula. The Socializing slice has empty prioritization.

(And functions are fixed, globally, in this example). The update can be serialized into a base update and a default update that are thus each strongly sympathetic. Therefore, by Theorem 6.29, the update is monotonic with respect to the previous Socializing slice. And, by the overall disjoint-sub-languages clean decomposition, it is thus globally monotonic.

This yet again illustrates the **synergy** created by combining different kinds of special-case decompositions, to get more locality, e.g., clean decomposition, than is available from any one alone.

This example illustrates another important strategic point. Here, a new special case (strong sympathy) "stepped into the breach" left behind by the loss of the sympathetically solitary property after the immediately previous update. This shows that when one cannot apply a particular case, one can still often apply another, though it may yield a less strong locality result. E.g., in this example, there is no explicit solution, or even further (finer-grain) clean decomposition, available (directly) from our results: the degree of guaranteed locality is weaker.

Reformulation Given the Base:
The forming constraint is quite restrictive. We observe that strong sympathy is especially applicable when it is combined with another kind of monotonic-logical equivalence-preserving reformulation.

Theorem 6.32 (Reformulation of Defaults Given the Base)
In any PDC:
Equivalence given the base suffices for reformulation of defaults into their equivalents.
More precisely: In $PDC(B; D; R; fix\ F; Z)$:
Suppose the base B implies the equivalence of a default formula $Di[Z, x]$ to some other formula $Ei[Z, x]$.

$$B[Z] \quad \models \quad \forall x. \, (Ei[Z, x] \equiv Di[Z, x])$$

Then the default whose formula is Di can forever-equivalently have that formula replaced by Ei.

I.e., let G be the result of replacing Di in the tuple D by Ei. Then

$$PDC(B; G; R; fix\ F; Z) \quad \equiv \quad PDC(B; D; R; fix\ F; Z)$$

Note that the new, reformulated default has the same index and prioritization as the old.

Proof : In the augmentation part of the original circumscription, $B[Z]$ implies that $Ei[Z, x]$ can be substituted for $Di[Z, x]$, and $B[Z']$ (inside the scope of the second-order quantifier) implies that $Ei[Z', x]$ can be substituted for $Di[Z', x]$. Thus the default pre-order for the new default is equivalent to the default pre-order for the original default, in the context of the circumscriptive maximization. □

Example 6.33 (Socializing, Strong Sympathy, Reformulation)

Continuing Example 6.31:
There, the base implies the monotonic conclusions that

$polite(Peg, Maggie)$

$listens(Peg, Maggie)$

Next, suppose there is another update $\mathcal{U}_{17}$ about Socializing:

$\bullet\!>\ understands(Peg, Maggie)$

The previous monotonic conclusion $listens(Peg, Maggie)$ can be exploited. It implies that this new base update's formula is equivalent to an instantiation of a previous default: the default that listening implies understanding ($s7$). Therefore, by Theorem 6.29 together with this **reformulation given the base** (Theorem 6.32): the update is monotonic with respect to the previous Socializing slice. And, by the overall disjoint-sub-languages clean decomposition, it is thus globally monotonic.

Detecting Forming May Be Difficult:

The forming conditions in our strong sympathy results are *defined* syntactically, but because they involve manipulation of whole (default and fixture) formulas, and (in general, unconstrained) logical equivalence, they may be difficult to detect from surface syntax.

This difficulty arises even more when performing reformulation, of course.

However, some kinds of forming, e.g., instantiation, and some kinds of logical equivalence, e.g., to resolve away a single ground literal, are relatively simple to perform. The above examples only involved such.

Chapter 7

Designing Reasoning Procedures

In this chapter, we discuss how to exploit our results to design procedures for NM specification, inference and belief revision: primarily, in prioritized predicate and default circumscription; and, more generally but to a lesser extent, in other NM formalisms as well. We sketch a new "Genealogical" architectural method for sound (but perhaps incomplete) NMR, e.g., learning agents in the sense discussed in sections 1.3 and 1.6. This architecture is able to exploit results on decompositions and safeties, e.g., ours in chapters 2 through 6. The architecture employs a novel kind of truth maintenance mechanism: we develop this in detail. This Genealogical Truth Maintenance System is an unusual variant on de Kleer's [1986a] Assumption-based TMS: each assumption corresponds to an entire global NM theory or an entire decomposed NM sub-theory. We call the architecture, and its use of truth maintenance, "genealogical" because they revolve around the use of sub-theory relationships: conclusions "descend" to the current global theory from slices, tiers, and previous theories. We show that the Genealogical Truth Maintenance task is computationally tractable: polynomial in the justifications (justifications in the sense of de Kleer), unlike the ordinary ATMS task, which is NP-hard.

We also sketch how to implement most of our major results on decompositions, safeties, specification, reformulations, and inference procedures from chapters 2 through 6. We observe that the computational cost involved, both to recognize the relevant conditions / special cases and to perform the decompositions etc., is polynomial in the size of the global CLD axiom set. Thus, we observe that our Genealogical NMR architecture is practically implementable.

Guide to Reader: Sections 7.1 through 7.3, taken together, provide a summary / overview of the whole chapter.

7.1 Introduction

In this chapter, we discuss how to exploit our results to design procedures for NM specification, inference and belief revision: primarily, in prioritized predicate and default circumscription; and, more generally but to a lesser extent, in other NM formalisms as well.

7.1.1 Challenges Addressed and Previous Incapabilities

Guide to Reader: Recall the discussion in section 4.2, especially of previous incapabilities of NMR (sub-section 4.2.1)

We address challenges of three kinds for circumscriptive reasoning procedures, in this chapter. We will show how our results can help with all of them, to overcome or mitigate the incapabilities of previous circumscriptive reasoning procedures.

(Our focus, as throughout this paper, is on relatively expressively rich circumscription. By expressively rich here, we mean beyond the relatively simple cases in which circumscription is equivalent to the Closed World Assumption, predicate completion, or stratified logic programs with negation; see section 8.7 for more discussion of these cases.)

1. There do not exist previously[1] any procedures for several basic reasoning **tasks**. These tasks are especially important for **learning agents**.
   (a) **Forward inference**: Neither exhaustive (complete) nor, even more importantly, **selective** (incomplete but focussed).
   (b) **Belief revision**: Neither complete, nor selective, with respect to a current set of conclusions (that is a proper subset of the current global theory).
   (c) **Specification**: No procedure, indeed not even a formal method, for **incremental** specification of the **policy**: i.e., the defaults / minimized predicates, their **precedence** (prioritization) p.o., and fixtures.
2. There do not exist previously any procedures for several **expressive** classes.
   (a) **Non-layered** prioritization: neither PPC nor PDC. (This is not surprising since we were the first to define it).
   (b) **Default**: Neither maximized-default, nor minimized-(elementary)formula, circumscriptions, especially for the case of non-layered prioritization.[2]
   (c) **Beyond propositional**: No base language that is not essentially expressively reducible to propositional. I.e., no first-order expressive class, e.g., **universal**, without the restrictions of ground or domain closure (see section 7.4 for precise details of these restrictions).

   We showed the importance for applications, e.g., learning agents and others, of the expressive dimension of non-layered prioritization in chapters 1 through 3. Likewise, though less critically, we showed, in chapters 1 through 3, the convenience and natural-ness of direct specification in terms of formula defaults.

[1] here and below: to our knowledge, in the literature

[2] But the procedures of [Ginsberg, 1989] and [Baker and Ginsberg, 1989] are easily shown to handle correctly (parallel and layered, respectively) PDC, not just PPC. See discussion in section 7.4.

3. Prioritized default circumscription, even in the parallel, propositional, predicate case, is intrinsically **computationally difficult**. Recall from sub-sections 4.2.2 and 4.3.1 that the *non-monotonic aspect* of entailment in parallel propositional PPC is (co-)NP-hard, beyond just the NP-hardness of monotonic entailment in the base-language propositional (classical) logic. Recall from sub-section 4.3.2 and section 4.4 that this computational difficulty can be viewed in terms of the *logical globality* of NMR, e.g., of conflict between defaults. To deal with this computational difficulty, it is desirable to pursue several tactics, especially for learning agents.

   (a) **Exploit special cases**: To recognize and take advantage of special cases of axiom sets or updates or conclusion blocs, that can be reasoned about relatively easily. An agent might be designed to operate only under expressive etc. special-case restrictions. More generally, however, we would like an agent to be equipped with a bag of tricks and the smarts to know when (and how) to use them.

   (b) **Reason incrementally but soundly**: live with incompleteness, without sacrificing correctness with respect to declarative semantics.

   (c) **Compute concurrently:** e.g., in inference and belief revision.

   There are no previous reasoning procedures for expressively-rich circumscription that employ any of these tactics. (Other than the fact that there are some relatively expressively-simple special-case procedures.)

These three kinds of challenges interact, of course. The difficulty of reasoning tasks is compounded by increased expressiveness: both to design a procedure at all, and to do so computationally efficiently.

7.1.2 How We Help

Our results on circumscription have uses of many different kinds for designing reasoning procedures. Next, we outline some of the most interesting, immediate, and concrete uses that respond to the challenges discussed in the last sub-section. We discuss these in this chapter; we defer discussion of other uses until sections 8.1 and 8.2.

1. **Default case via predicate case:** Our results on abnormality theories (sub-section 3.3.2) imply how to expressively generalize any previous or future inference (or belief revision) procedure for (a subclass of) prioritized *predicate* circumscription so as to apply it to prioritized *default* circumscription. E.g., the previous query-answering procedures for the predicate case can all be generalized straightforwardly to the default case.

2. **Incrementally specify policy, e.g., defaults and prioritization:** Our results on defining CLD (section 3.2) imply how procedurally to specify any prioritized predicate circumscription (PPC) or prioritized default circumscription (PDC), including with *non*-layered prioritization

p.o., *incrementally* via CLD. That is, they imply how to dynamically *update* a PPC or PDC in terms of adding new axioms, in such a way that the *policy*, not just the base, is updated in a modular / local fashion.

3. **New special-case inference procedures:** Our results imply new procedures for forward inference, incomplete (selective) and exhaustive (with respect to the NM aspect), in two special cases: directly, in totally-prioritized propositional (sub-section 5.13.2); and, less directly, in sympathetically solitary (section 6.5). For sympathetic-solitary, the inference is **tractable** (polynomial in the size of the CLD axiom set).

4. **Forward inference in non-layered case, via inference in parallel case:** Our results on canonical decomposition along prioritization (chapter 5, especially section 5.4) imply how to procedurally perform (sound but incomplete) forward inference in *non*-layered PPC / PDC via (sound but incomplete) inference in parallel PPC / PDC; and to perform *exhaustive* (sound) forward inference in non-layered via exhaustive (sound) forward inference in parallel. Remember that there are previous inference algorithms available for (essentially propositional) parallel PPC, but not for non-layered PPC, nor for even parallel PDC.

5. **Selectiveness and concurrency in forward inference:** Our results imply how to procedurally perform forward inference selectively and concurrently: by combining (especially conjunctive) decomposition of several kinds with any exhaustive procedures available for special cases. E.g., exhaustive (concurrent) inference on slices that are totally-prioritized propositional or sympathetically solitary can be combined conjunctively to yield forward inference that is selective from the global viewpoint.

6. **Rapidly localize belief revision:** Our results imply how to procedurally perform selective parts (and, sometimes, all) of belief revision relatively rapidly: by automatically recognizing bloc safeties of updating for several special cases. These cases include: prioritization updates (sub-section 2.9.1), fixed updates and conclusions (sub-section 3.5.6), higher-priority conclusions (section 5.12), disjoint sub-languages (section 6.2), conservative extensions (section 6.4), and sympathetic updates (section 6.5).

7.2 A Genealogical Architecture for NMR and Learning Agents

In this sub-section, we sketch a new **"Genealogical"** NMR architecture (**GNMR**) **(terminology)** that combines all of the uses listed in sub-section 7.1.2, and responds to all of the challenges listed in sub-section 7.1.1. This GNMR architecture is a loose one; view it more as a **method or style category of architecture** rather than as a single specific architecture. By architecture, we mean a way of organizing a procedure or (sub-)agent.

The GNMR architecture is intended as the NMR component of a large-scale, expressively rich **learning agent** (recall sections 1.3 and 1.6). The learning agent, at any point in axiom update sequence, stores a "working" body of conclusions that is **sound**, but usually incomplete, with respect to the NM formalism. In learning agents, belief revision and safety are important issues, and where one wants to avoid the re-computation of inferences if one can do so without too much effort. However, many of the ideas and results used in the GNMR architecture, e.g., about decompositions and specification and reformulations, are applicable profitably to "static" NMR, i.e., to pure inference without (updating or) belief revision. Indeed, the learning agent NMR problem subsumes the static NMR problem.

The GNMR architecture performs (sound) inference, and belief revision, by exploiting decompositions and safeties (i.e., localities) and special-case inference procedures. In doing so, it uses truth maintenance.

The GNMR architecture uses as building blocks a **library** of various kinds of reasoning procedures. These library inputs include one or more procedures for each of: non-monotonic inference, monotonic inference (e.g., to perform conjunctive combination or to be called by the NM inference procedures), **meta-reasoning** about decompositions and safeties, and truth maintenance. (And, optionally, in addition, procedures that directly perform belief revision.)

The GNMR architecture is applicable both to circumscription and to other NM formalisms besides circumscription. The Genealogical **Circumscriptive** Reasoning architecture (**GCR**) **(terminology)** is the (prioritized-default) circumscriptive version of the GNMR architecture. In addition to inference and belief revision, the GCR architecture also (optionally) procedurally performs representation and reformulation reasoning tasks involved in *specification*: it maps between CLD and PDC and abnormality-style PPC. This can be viewed as another kind of meta-reasoning.

Building upon the library input procedures, the GNMR architecture yields as output an overall NMR procedure. This procedure takes as **dynamic inputs**:

1. the specification of the current global NM axiom set:
   a stream of axiom updates in the NM formalism; and

2. some specification of reasoning tasks: e.g., most typically,
   forward and backward inference requests for the current NM theory —
   the need for belief revision is (and may be left) implicit.

The reason that we call the GNMR architecture "genealogical" is that conclusions "descend" to the current global theory both from current sub-theories via decomposition, and from past sub-theories via safeties. A variety of general-case and special-case NM inference (and belief revision) procedures work on sub-theories. Meta-reasoning procedures recognize special cases, decompositions, and safeties. The **truth maintenance mechanism** performs the bookkeeping involved in tracking which conclusions proved (non-monotonically) for current and past sub-theories are IN (i.e., provable non-monotonically) for the current global theory. Thus, we call its truth maintenance mechanism "genealogical", as well. The most basic function of the truth maintenance mechanism is to store the the current working conclusions of the learning agent.

The GCR / GNMR architecture is unimplemented. And, in this paper, we only sketch it, e.g., many of its control aspects. Much work remains to flesh out its vision. In the rest of the chapter, we elaborate only partially on the overall GCR / GNMR architecture: bottom-up as we detail its elements. (See especially sub-sections 7.5.2 through 7.5.5, 7.7.1 and 7.7.2.) We show, however, that all of the key elements of the GCR architecture are **implementable**, and that all of the key elements besides the basic circumscriptive theorem-proving (i.e., inference local to the sub-theories) can be performed computationally tractably, i.e., worst-case **in polynomial time**, for interesting, expressively rich cases. And the main point of the architecture is to cope with the (worst-case intractable) computational difficulty of circumscriptive NM inference (and belief revision). Thus the GCR architecture can tackle that difficulty, while imposing only reasonable (i.e., tractable) overhead.

7.3 Summary of Rest of Chapter

The GCR architecture can use, as a procedural piece, any (sound) circumscriptive inference procedure. That is, it takes circumscriptive inference procedures as library inputs. The capabilities of the GCR architecture thus "piggyback" on those of the available circumscriptive inference procedures.

In section 7.4, we accordingly **review the previous inference procedures available** for (especially expressively rich) circumscription. These are capable of query-answering (i.e., backward inference) in the entire class of: predicate circumscriptions with layered priority whose base languages are essentially reducible to propositional; i.e., layered, essentially-propositional PPC. These previous inference procedures thus provide a fairly nutritious grist for our GCR mill. In addition, our analysis of their expressive limitations helps to define the frontier, discussed in sub-section 7.1.1, relative to which we are able to demonstrate progress.

In section 7.5, we detail the **(terminology) Genealogical Truth Maintenance** (**GTM**) architectural task and **System**. We show that the GTM task is tractable, and sketch algorithms for it. The **GTMS** is a kind of Assumption-based Truth Maintenance System (ATMS) cf. de Kleer [1986a]. It is a novel and unusual application of the (A)TMS idea in several regards. The role of an assumption in an ATMS is played in the GTMS by an entire NM theory, e.g., a second-order-logical circumscription formula. The GTMS is more complicated in one respect than an ordinary ATMS: assumptions may be consequents in justifications: super-theories may imply sub-theories. But the GTMS is simpler in another respect: only single-assumption environments (i.e., single NM theories) are of interest for queries and labelling within the GTMS. This simplicity ensures tractability: to ensure that queries cost constant time, the GTMS maintenance / updating costs time that is (worst-case) **polynomial** in the size of the justifications stored by the GTMS. By contrast, ordinary ATMS maintenance / updating is NP-hard.

Truth maintenance systems have been used previously in support of NM inference. However, our style of application is unusual. We are not using justifications that are themselves non-monotonic in the style of Doyle [1979], i.e., that may have OUT (his terminology) antecedents. And we are not, in any sense, using the truth maintenance mechanism in the mode of abduction, in the style of previous applications of the ATMS to support NMR, e.g., [de Kleer, 1986b] [Ginsberg, 1989] [Selman and Levesque, 1990] [Junker, 1991] [Raiman and de Kleer, 1992]. Rather, we are using the truth maintenance mechanism as a bookkeeper, in a fundamentally "monotonic" fashion.

In section 7.6, we discuss the specification aspect of the GCR architecture. We sketch how to **mechanize CLD as a specification tool** for PDC or PPC. One aspect of this specification reasoning is that the information in the (current global) prioritization axioms in CLD has to be transitively closed to generate the (current global) prioritization partial order. We show how to perform this transitive closure efficiently: each prioritization update costs (worst-case) time **linear** in the number of default axioms. (Other CLD updates cost constant time.) Another aspect of specification reasoning is reformulation between PDC and abnormality-style PPC. Reformulating from PDC to PPC is useful to exploit inference procedures available for PPC. Reformulating from PPC to PDC is useful to exploit meta-reasoning about decompositions and safeties. We show how to perform these **reformulations** efficiently: in time that is (worst-case) **linear** in the size of the (CLD) axiom set.

In section 7.7, we sketch how to procedurally **implement most of our major results on decompositions, safeties, and inference procedures from chapters 2 through 6**. We show that the computational cost involved, both to recognize the relevant conditions / special cases and to perform the decompositions etc., is (worst-case) time **polynomial** in the size of the global CLD axiom set.

7.4 Previous Inference Procedures for Circumscription

Guide to Reader: Recall the discussion in sub-section 7.1.1 of previous incapabilities of circumscriptive reasoning procedures.

In this section, we discuss previous inference procedures for circumscription. We **focus on the expressive classes** they handle, rather than on the details of how they work. **Recall the summary of this section** in the beginning of section 7.3.

In several relatively expressively simple cases, circumscription corresponds to by now well-understood and widely-used NM inference mechanisms that were originally developed without reference to circumscription. E.g., special cases of layered-prioritized predicate circumscription correspond to: the Closed World Assumption (see [Gelfond *et al.*, 1989] which reviews and elaborates this relationship); a broad special case of predicate completion cf. [Clark, 1978] (see discussion of the solitary case in the beginning of our section 6.5); and Prolog-style stratified logic programs with negation (see [Lifschitz, 1987a] [Przymusinski, 1988]). Note that, in all of these cases, there is no conflict between defaults (minimized predicates), and thus the non-monotonic aspect of inference is rather simple computationally.

For **expressively rich (terminology)** circumscription, i.e., beyond these relatively simple special cases of layered PPC, there are no previous procedures[3] for forward inference or belief revision, only for query-answering: [Przymusinski, 1989] [Ginsberg, 1989] [Baker and Ginsberg, 1989] [Inoue and Helft, 1990] [Helft *et al.*, 1991].

Observation 7.1 (Expressive Restrictions in Previous Procedures)
The previous query-answering procedures (listed above) for circumscription have only been proved correct (sound and complete) under the imposition of several **expressive restrictions** on the circumscription, beyond the base language (hence, the base and the query) being **first-order**:

1. The prioritization partial order is **layered**.

2. **Predicates** are minimized, rather than (elementary-)formulas being minimized or default formulas being maximized. (But see discussion below of Ginsberg's [1989] and Baker and Ginsberg's [1989] procedures, which are easily shown to extend to the default case.)

3. All **function** symbols are **fixed** (as McCarthy [1980] [1986] and Lifschitz [1985] do in their definitions of predicate circumscription and layered-prioritized predicate circumscription, respectively; and as we assumed on page 36).

4. The base contains (or implies) **the uniqueness of names** (Definition 2.21).

5. EITHER:

   (a) The base and query are **ground**; OR

   (b) The base contains (or implies) **domain closure** (Definition 2.23); and the base language (hence, the base and the query) has **no function** symbols of arity greater than zero.

Next, we give some analysis of these expressive restrictions.

Propositionality:
As Gelfond *et al* ([1989], page 84) remark, (first-order plus) domain closure and the uniqueness of names and no functions of non-zero arity, taken together, imply that: any base sentence (closed formula) is expressively reducible to a propositional combination of ground atoms. In our terms, we **observe**, moreover, that they imply that: (**terminology**) the base language (hence, also, queries in it) is **effectively propositional.** Any universally-quantified sentence (e.g., clause) in the base language can then be viewed as a schema standing for the (finite) conjunction of all its ground instantiations.

Likewise, the ground case (especially with unique names) is essentially propositional: the domain is essentially closed to those individuals named in: the base plus the query. The circumscription with a domain closure axiom added is then equivalent, for purposes of answering (ground) queries, to the circumscription without it.[4] Note that domain closure and unique names imply that any first-order formula is expressively reducible (equivalent) to a ground formula.

[3] in the literature, to our knowledge

[4] This remark can be made more precise and rigorous. We do not take the space to do so.

It is not too suprising that all of these previous procedures restrict the domain in such a way as to make things essentially propositional: even for monotonic inference in (full) first-order logic, one cannot expect completeness in an algorithm. Propositionality ensures decidability.

Implications Among the Restrictions:
Domain closure and uniqueness of names together imply that the set of ground terms is finite. We **observe** that, therefore, when one imposes the restrictions of domain closure and uniqueness of names, then one typically needs / wants also to impose the restriction that there are no functions with non-zero arity.

More precisely: If there are functions of non-zero arity, then (in ordinary first-order predicate calculus) there are an infinite number of ground terms, since any function f with non-zero arity m may be applied (recursively) to a tuple of ground terms $\langle t1, \ldots, ti \rangle$ to produce a new ground term: $f(t1, \ldots, tm)$.[5] One might define functions in such a way as to be restricted in their applicability over the domain, but this involves formal complications, e.g., types, sorts, or partial functions, etc..

In any event, one can always rewrite (i.e., definitionally reformulate) a finite set of ground terms: to make each instead into an individual constant, i.e., into a zero-arity function symbol.

Recall that, by Theorem 2.24, domain closure and the uniqueness of names together imply that all functions are indirectly fixed (Definition 3.47), even if they are not directly fixed.

We **observe**, therefore, that the **most essential expressive restrictions** among the above-listed **in Observation 7.1 are** thus: **layered, predicate, domain closure, and uniqueness of names**.

Satisfiability and Well-Behavior:
Recall also that, by Theorem 3.24, domain closure plus the fixing of functions (e.g., a complete theory of equality) implies that the circumscription is quasi-propositional, and, therefore, well-behaved: it is well-founded and "well-founded"; it preserves satisfiability; it obeys immunity of the fixed (by Theorem 3.39); and it is cumulative (see sub-section 4.8.1).

Next, we discuss each of these previous procedures in a bit more detail.

Przymusinski:
Przymusinski [1989] develops a procedure by drawing, in great part, on his joint work with Gelfond and Przymusinska ([Gelfond *et al.*, 1989]) relating circumscription to what they call the "extended" closed world assumption. These latter results in effect gave some proof theory for circumscription: they reduced entailment problems in circumscription to entailment problems in (the base-language) first-order logic, via negation as failure.

Przymusinski gives a sound and complete (decidable) algorithm that requires several **additional expressive restrictions** besides those listed in Observation 7.1:

- The base and query are **ground**.

[5] unless there are too few functions of zero arity to get this recursion started; i.e., unless there are fewer individual constants than the smallest arity of any function with non-zero arity

- The base and query do **not mention equality** (other than, in effect, in the uniqueness of names[6]).
- **Queries are closed** formulas, i.e., sentences.
- The prioritization partial order is empty, i.e., **parallel**.

Przymusinski describes briefly, further, an extension of this procedure to the **non-ground, universal (clausal)** case. However, he does not fully detail the extended procedure or the expressive restrictions under which it is correct; and he does not give any proof of its correctness. He also gives, briefly, an extension of the procedure to the case of **layered prioritization**: by recursively using the parallel-case procedure on each layer. Again, he does not give a proof of its correctness.

Ginsberg; Baker and Ginsberg:
Baker and Ginsberg [1989] give a sound and complete algorithm for **layered prioritization**. This extends an earlier algorithm by Ginsberg [1989] for the parallel case, based on a backward-chaining assumption-based truth maintenance system (ATMS) [de Kleer, 1986a]. Baker and Ginsberg's method deals with priorities via a kind of dialectical argumentation, involving rebuttal and refutation. Like Ginsberg, they are able to exploit an arbitrary first-order theorem-prover. By contrast, Przymusinski's method is based on a particular kind: ordered linear resolution.

Baker and Ginsberg impose several **additional expressive restrictions** besides those we listed above in Observation 7.1:

- **Domain closure** and no functions with non-zero arity.
- There are **no fixed predicates**.
- **Queries are closed** formulas, i.e., sentences.

As Baker and Ginsberg have observed in private communication (spring 1989), their method is straightforwardly extended, however, to permit fixed predicates: by using Corollary 3.44.

Default Case: We **observe** that both Ginsberg's procedure and Baker and Ginsberg's procedure are essentially developed, conceptually as well as mathematically, in terms of maximizing default pre-orders on models. Thus they are easily extended to handle correctly the default, not just the predicate, case.

Inoue and Helft:
Inoue and Helft [1990] analyze Przymusinski's procedure for the **parallel** case of priorities, and Ginsberg's algorithm (which, remember, is restricted to the parallel case). They compare them, show how to unify them, and show how to improve their efficiency. As part of unifying the two, they show (their proposition 2.10) how to extend Ginsberg's algorithm so as to permit fixed predicates. We **observe** that their result is very close, i.e., can be viewed as essentially equivalent, to applying Corollary 3.44 (as discussed above in connection with Baker and Ginsberg's procedure).

[6] and perhaps in domain closure: see discussion of non-ground case below

Inoue and Helft impose the same expressive restrictions as Przymusinski does for the parallel case.

Apparent confusion: Inoue and Helft say (beginning of their section 2) that Ginsberg's imposition of the domain closure restriction is "unnecessary", since: 1) they show how to unify Ginsberg's algorithm with Przymusinski's; and 2) Przymusinski "claims ... [his] algorithm [is] applicable to the first-order case with [uniqueness of names] ...". However, as we observed earlier in connection with Przymusinski's procedure, there seems to be a confusion here: Przymusinski's results guaranteeing the *correctness* of his algorithm assume the ground case, and, in any event, apparently use the restriction of domain closure in Gelfond *et al*'s results. E.g., Przymusinski's theorems 2.5 and 2.6, which Inoue and Helft analyze, are taken directly from Gelfond *et al*, who impose domain closure. Also, as we remarked above, the ground restriction is very close to domain closure. Thus, it does not seem quite appropriate to say that domain closure is "unnecessary" without further elaboration.

Cumulativity and inherent limitations of approach: We **observe** that Inoue and Helft's approach, like Przymusinski's approach, assumes the cumulativity (see sub-section 4.8.1) of the circumscription. Thus their basic results (e.g., their Theorem 2.7) guaranteeing correctness of their procedures fail to hold when the circumscription may be non-"well-founded" (Definition 3.16).

Opportunity to apply to layered-prioritized case: We **remark** that Inoue and Helft's generalizations and improvements of Przymusinski's procedure for the parallel case, and of Ginsberg's algorithm, suggest the opportunity to generalize and improve both Przymusinski's procedure for the layered-prioritized case, and Baker and Ginsberg's algorithm for the layered-prioritized case.

Helft, Inoue, and Poole:
Helft *et al* [1991] extend Inoue and Helft's ([1990]) procedure so as to deal with open queries with answer extraction, not just closed (yes/no) queries.

7.5 Genealogical Truth Maintenance

Guide to Reader: We assume familiarity with the basic ideas of truth maintenance [Doyle, 1979] and its assumption-based variant [de Kleer, 1986a]. Throughout this section, we use **de Kleer's terminology** in describing the ATMS and GTMS ideas.

In this section, we develop a new kind of truth maintenance mechanism: the Genealogical Truth Maintenance System (GTMS). **Recall the earlier overview discussion of truth maintenance** in sections 7.2 (towards the end) and 7.3 (in the middle).

7.5.1 Definition of the GTMS

Definition 7.2 (Genealogical Truth Maintenance System)

The Genealogical Truth Maintenance System (GTMS) is a **strict special case** of de Kleer's [1986a] Assumption-based Truth Maintenance System (ATMS). Relative to the ATMS, the GTMS is defined by one specializing **restriction**:

1. **Single-assumption environments**: In queries to the GTMS, and in labels within the GTMS, each environment contains at most one assumption.[7] In particular, the completeness of the GTMS is only guaranteed for such queries and labels.

ATMS Terminology: Here, and throughout the remainder of this chapter, we use de Kleer's ATMS terminology: e.g., "assumptions", "environments", "nogoods", "labels", "queries", "justifications", "nodes", and "problem-solver".

In the remainder of this section, we describe the GTMS primarily in terms of its application to the GNMR architecture. We discuss what role the GTMS fills within the GNMR architecture, why we designed / defined it as we do above, and how it is applied to support the GNMR architecture. However, the **GTMS is potentially applicable beyond the GNMR architecture**. E.g., the justifications need not be used for representing NM inferences and meta-inferences.

7.5.2 Bookkeeping Task in the GNMR Architecture

The GTMS occupies one of the central roles in the GNMR architecture. The task of the GTMS is a kind of **bookkeeping** in support of exploiting decompositions and safeties. In this sub-section, we describe this bookkeeping task. In detailing the GTMS, we thus elaborate on the sketch, given earlier in section 7.2, of the GNMR architecture.

The most basic function of the truth maintenance mechanism is to store the current working conclusions of the learning agent. More generally, the bookkeeping task involves a limited amount of inference and belief revision, that complements the reasoning performed by the other procedures in the GNMR architecture. The bookkeeping reasoning can be viewed as involving meta-reasoning about NM theories.

In the GNMR architecture, NM inferences are made by one or more NM inference procedures. Each NM inference[8] is made by one particular NM inference procedure working on one particular axiom set / theory. Thus considering the overall set of NM inferences made, over the course of the sequence of axiom updates up to the presently-current in a learning agent, there are several kinds of NM inferences made while working on many different axiom sets / theories:

- NM inferences made in past global axiom sets
- NM inferences made in past decomposed axiom sets: slices and tiers;
  and, remember, there may be a hierarchy of decomposition: slices within slices, etc.

[7] **Additional, clarifying detail:** We view each nogood as an environment in the label for the contradiction node $\perp$ which denotes False. A justification may have the contradiction node as its consequent but not as one of its antecedents. The contradiction node may not be an assumption.

[8] Here, and throughout this section, keep in mind that monotonic inference is a special case of NM inference.

- past monotonic inferences, e.g., within a NM theory, or in the process of performing (partial) conjunctive combination of slices, or, more generally, as part of the process of NM inference but not representing a sanctioned conclusion
- NM inferences made in the presently-current global axiom set
- NM inferences made in the presently-current decomposed axiom set
- presently-current monotonic inferences

In the GNMR architecture there are, in addition, procedures that perform meta-reasoning to find decompositions and safeties, e.g., by applying our results in chapters 2 through 6 (see section 7.7 for discussion of how this can be done). Thus considering the decompositions and safeties found / established over the course of a sequence of axiom updates in a learning agent, there are several kinds of **"meta-inferences" (terminology)** about decomposition and safety relationships among many different axiom sets / theories. These decomposition and safety relationships are about (i.e., mention) axiom sets / theories both past and presently-current, and both global and decomposed.

- decomposition:
  - conjunctive
  - serial
- safety:
  - preservation under the then-latest update, of
the immediately-previous "global" theory,
via application of a global monotonicity result
(actually, "global" here might be itself be already-decomposed)
  - preservation under the then-latest update, of
one or more immediately-previous decomposed sub-theories (more precisely, preservation of their conjunctive combination if there are more than one such slices),
via application of differential decomposition results
  - global redundancy, i.e., equivalence after the then-latest update.

Terminology: In the GNMR architecture, by "**genealogy**", we mean **ancestral relationships** between NM theories, in the following sense. Relative to one NM theory a bunch of other NM theories may be *ancestors* to it, in the sense that their conclusions carry over, i.e., *descend*, to it. For any particular "given" (e.g., "current") global NM theory, the genealogy tells how conclusions in that theory *descend* (i.e., carry over) from a bunch of other NM theories: 1) its historical forebears; and 2) its decomposed pieces. These historical forebears are past points ("moments") in a sequence of axiom updates: that is, they are previous global axiom sets' NM theories. Safeties of updating imply that much of those historical forebears' conclusions descend, unrevised, into the given NM theory. The decomposed pieces, by contrast, do not (necessarily) correspond to past historical moments. Decompositions imply that their conclusions, and conclusions based on purely monotonic reasoning from their conclusions, descend into the given NM theory. There is an interesting synergy between these two kinds of "ancestral descent" of conclusions. A piece theory may itself have conclusions that descend from historical forebears. And a historical forebear may itself have ("historical") pieces. The genealogy represents this information not just for one particular given NM theory, but for many NM theories (each of which may be considered as the "given" focus of attention).

With the above NM inferences and genealogical meta-inferences as inputs, the GTM task is to support the exploitation of decompositions and safeties by the GNMR architecture; i.e., to support at each point in the axiom update sequence:

- composition of theories (i.e., the inverse of decomposition): to have conclusions **"descend" (terminology)** to the global (axiom set / theory) from each
  - slice
  - tier
  - conjunctive combination of any subset of slices

  ; and to do so hierarchically: i.e., to have conclusions in the j^{th} slice of a slice i descend to the slice i and then descend to the global, etc.. Note that *every* conclusion in a slice or tier or conjunctive combination of slices should descend to the global (and ditto hierarchically).
- safety: to have (known-to-be-)preserved conclusions descend
  - to the presently-current global
    from the immediately-previous global
  - to a current slice or tier or conjunctive combination of slices
    from its immediately-previous version (i.e., correspondent in a global monotonicity or differential decomposition result)
  - to the presently-current global or slice or tier or conjunctive combination of slices
    from any past version, not just the immediately-previous version;
    i.e., to descend **"historically" (terminology)** in a transitive manner over the sequence of axiom updates
  - to the immediately-previous global from the presently-current global ; more generally, to descend against the flow of history, via redundancy, in a transitive manner
- seamless synergy between all of the above kinds of descent, especially between (de)composition and safety: e.g.,
  to have conclusions descend to the presently-current global from the conjunctive combination of two slices from different points in the past

More precisely, by "descend" above, we mean that conclusions which are known to be sanctioned for the "from" axiom set / theory, e.g., a slice, should also be known to be sanctioned for the "to" axiom set / theory, e.g., the global theory to which that slice belongs.

The GTM task, moreover, is to keep all this belief information in a readily-accessible, up-to-date form: so as to support the learning agent's actions based on its beliefs, as well as its NMR. The most basic aspect of the GTM task is to store the learning agent's current working set of NM conclusions. By "belief", both here and throughout this thesis in the context of discussing GTM, we mean a sentence that is **known to be entailed** in the NM logical formalism, by the agent's current global axiom set. By "known to be" and "up-to-date" here, we mean with respect to the currently available set of NM inferences and meta-inferences that have already been made elsewhere in the GNMR architecture and have been inputted to the GTM (sub-)System (GTMS). The agent needs to be able to retrieve its current beliefs quickly, both in the course of acting (or deciding how to act) and in the course of performing further NM inference or belief revision (or specification). By "quickly" here, we mean essentially by lookup: more precisely, with computational cost per believed sentece that is (worst-case) constant or log time, as in a database.

For action, i.e., the learning agent's "performance", typical queries to the GTMS are to ask:

- Is sentence pi (known to be) believed in the current global axiom set / theory tj?
  Notation: We will write this as: ?IN(pi,tj).
  This essentially corresponds to the usual usage of **IN** in the truth maintenance literature (e.g., Doyle's original paper [1979]). The main difference is that we make the theory context tj explicit.

- What are all the sentences p (known to be) believed in the current global axiom set / theory tj?
  I.e., IN(?p,tj).

For the agent's continuing NM inference (by various procedures within the GNMR architecture), both backward and forward, typical queries to the GTMS are also to ask the above, and, in addition, perhaps to ask:

- Which are the current slices or tiers t, i.e., of the current global axiom set / theory tj?
  I.e., IN(?t,tj).

- In which axiom sets / theories t is sentence pi (known to be) believed?
  — e.g., so as then to go and to work on them, during focussed forward inference.
  I.e., IN(pi,?t).[9]

However, the GTM operations beyond just these retrievals should also not themselves impose too high an overhead relative to the other computational costs in the GNMR architecture. E.g., the GTM computation should not dominate the overall GNMR architecture's computation.

[9]In answering such a query, the reply might be the True enviroment. This may be denoted, for example, by the node $\top$.

7.5.3 Design Rationale: GTMS Functionality for GNMR

In this sub-section, we explain the design rationale behind the GTMS: i.e., why the GTMS functionality is appropriate for the bookkeeping task in the GNMR architecture.

Given the definition of the GTM task in the last sub-section, why should the GTM System indeed be a truth maintenance system, as opposed to some other kind of system?

In the GTMS task:

1. One needs to make use of justificational **dependency** information: e.g.,
   (a) the dependence of a NM conclusion (sentence) pi on the axiom set / theory tj in which it was inferred; and
   (b) the dependence of a monotonic conclusion (sentence) pi on the antecedents (sentences) $q1, \ldots, qm$ from which it was inferred.
2. One is working in a changing context, i.e., the current global NM theory, such that sentences that were previously justified may be no longer.
3. One wants to re-use past (inferential) work: to avoid re-computing from scratch.

The concept of truth maintenance ([Doyle, 1979]) is designed to deal with these three attributes.

The GTMS task has several more attributes, however, which are difficult for a Doyle-style TMS to handle:

4. One needs to have, and work in, multiple contexts (NM theories) at once, not just a single current context: i.e., the NM (sub-)theories, arising from decomposition and safety, that are involved in descent to the current global NM theory.
5. One wants to be able to move between these multiple contexts easily, so as to work in several current sub-theories at once or in rapid succession; e.g., in support of concurrent NM inference processes that query the GTMS, or as axiom updates are received.
6. One wants to keep all the work that has been done in all contexts. Even if a given past (sub-)theory is not involved in descent to the current global NM theory, nevertheless, it may in the future become so involved in descent, e.g., as:
   (a) the global axiom set changes; and/or
   (b) the GTMS receives an additional meta-inferences that establish new decomposition or safety relationships.
7. Yet, the different contexts (theories) may be mutually contradictory / inconsistent. (After all, the potential for such inconsistency is the whole point of logical *non*-monotonicity.)

The concept of *assumption-based* truth maintenance ([de Kleer, 1986a]) is designed to deal with these additional four attributes. Accordingly, the first step of our functional design is that:

The GTMS is a special case of an ATMS.

But how is one to represent NM theories and genealogical meta-inferences about decompositions and safeties within the ATMS framework? Our solution is as follows.

Assumption = NM Theory:
We represent a NM theory as an assumption. That is, the role played by an **assumption** in a standard, ordinary Assumption-based TMS (ATMS) [de Kleer, 1986a] is played in the GTMS by the **name of a NM theory**. **Terminology:** Henceforth, in speaking of the GTMS, we will often say simply **"NM theory"**, or **"theory"**, instead of "name of NM theory". Also, for purposes of comparative exposition, we will sometimes refer to (names of NM) theories as "assumptions" in the sense of a standard ATMS.

In chapters 1, 2, and 3, we defined a theory as a set of conclusions in some logical language. One can view this set of conclusions as a (possibly infinitary) sentence. In circumscription, the theory can be identified (essentially, by equivalence in higher-order logic) with the circumscription formula sentence. (Note that in the case of circumscription, this sentence is always finitary.)

Genealogy Via Sub-Theory Relationships:
We represent the genealogical meta-inferences by extracting from them only the information that is relevant to the GTMS. Regardless of whether the meta-inference is about conjunctive or serial decomposition, or partial or global or redudant safety, we represent it in one uniform fashion: as a set of one or more sub-theory relationship statements. By sub-theory relationship, we mean that one (axiom set's) theory is a sub-theory of a second.

Rather elegantly, these sub-theory relationships provide *exactly* (i.e., no more and no less than) the information that is needed to justify the ancestral descent, of conclusions from NM theories to other NM theories, that we discussed in the last sub-section. More precisely, in the ATMS framework, we represent each of these sub-theory relationship statements as a **genealogical justification** (see next sub-section).

Clearly, the sub-theory relationship is transitive. The GTMS computes, maintains, and uses the transitive closure of the sub-theory relationship statements; in the next sub-section, we discuss how. We call this transitive closed sub-theory relation: the **"GTMS genealogy" (terminology)**; more briefly, in the context of discussing the GTMS, we will call this simply "the genealogy".

7.5.4 Application of the GTMS to GNMR

In this sub-section, we describe the application of the GTMS to the GNMR architecture.

Review of ATMS Justifications:
Recall that an ATMS takes as its input the ongoing assertion of *justifications*. A justification

is a propositional Horn clause, with non-empty head[10] whose primitive propositional letters are *nodes*. Each node denotes a "problem-solving datum" in the words of de Kleer. Assumptions are a distinguished, special kind of node. We call non-assumption nodes: **ordinary (terminology)**.

In an ATMS, a justification is the record of an *inference step*, which *justifies* a **consequent** node, on the basis of, i.e., **dependent** on, zero or more **antecedent** nodes. A justification has the form:

$$a1, \ldots, an \quad \Rightarrow \quad c$$

where: $a1, \ldots, an$ are the antecedents, and c is the consequent, and the number n of antecedents may be 0, i.e., the left-hand-side may be empty.

Semantics: The ATMS is typically used in support of a problem-solver. In the context of the problem-solver, each node may have a complex meaning. Within the ATMS, however, a node is treated as opaque with respect to that meaning. That is, *within* the ATMS processing and responses to queries: each node is treated simply as a primitive propositional letter. A justification is interpreted as standing for a Horn clause (with non-empty head):

$$(a1 \wedge \ \ldots \ \wedge an) \supset c$$

In the case where $n = 0$, this is just the unit clause c.

Semantics of Nodes in terms of the GNMR Architecture:

In the GTMS, each ordinary node denotes a sentence in the base language. We also call an ordinary node a **proposition**. Each assumption node denotes a (NM) theory. (As we indicated earlier, we will also say "theory" for assumption.) In the GTMS's application to circumscription, i.e., in the GCR architecture, a theory is identified with a second-order-logical sentence: the circumscription formula.

Notation: Let " ' ", i.e., back-quote, stand for the **denotation** operator. Thus 'g stands for the thing named by g.

GTMS Justifications:

In the GTMS's application to the GNMR architecture, there are three basic kinds of justifications. (There may be other kinds, as well; however, we will not exploit that generality in this paper.)

Semantics: *Within* the GTMS, just as in an ordinary, standard ATMS, each justification, regardless of its kind, is treated as standing for a propositional Horn clause (with non-empty head). In the context of the GNMR architecture (i.e., the "problem-solver" in the ATMS sense), the nodes have more complex meanings: e.g., in its circumscriptive application, they stand for sentences in first-order or second-order logic. However, within the GMTS's processing and responses to queries, the nodes are treated as opaque with respect to that meaning. I.e., each justification is intepreted as a clause, just as we described above for ATMS justifications.

Next, we describe each **kind** of GTMS justification. After that, we describe each's meaning in terms of the GNMR architecture.

1. A type-1 justification represents a **NM inference**. This kind of justification has the form:

[10] A nogood can be viewed as having the contradiction (False) node $\perp$ for its head.

$t \quad \Rightarrow \quad p$

where t is a assumption node and p is an ordinary node. That is, there is exactly one antecedent, which is a theory, and the consequent is a proposition.

2. A type-2 justification represents a **monotonic inference**. This kind of justification has the form:

   $q1, \ldots, qn \quad \Rightarrow \quad p$

   where all the appearing nodes are ordinary, and n may be 0. That is, the antecedents and consequents are all propositions.

3. A type-3 justification represents a **genealogical meta-inference**. We also call it a **genealogical justification**. This kind of justification has the form:

   $ti \quad \Rightarrow \quad tj$

   where both the antecedent ti *and the consequent* tj are assumption nodes. That is, there is a single theory antecedent, and the consequent is also a theory.

Semantics of IN:

The answer $IN(p,t)$ by the GTMS, e.g., in response to a query to the GTMS, means (more precisely, necessitates) that the GTMS clause:

$`t \supset `p$

follows from the current set of justifications, i.e., is implied by the current set of GTMS clauses. Let J stand for this set, viewed as a conjunction of clauses. Then we can write this as:

$J \models `t \supset `p$

or

$\models J \wedge `t \supset `p$

In addition, the GTMS, like the ATMS, checks that t is not (known to be) inconsistent. I.e., $IN(p,t)$ is answered affirmatively if and only if: the above condition holds AND:

$J \not\models `t \supset \bot$

Semantics of justifications in terms of GNMR:

Next, we describe the semantics of the GTMS justifications in terms of what they mean in the context of the GNMR architecture.

1) A NM inference (i.e., type-1) justification is the record of an inference *within the antecedent theory*. The justification in this case is the record of a conclusion being made, in the **non-monotonic** logical formalism, from the theory's axiom set. A NM inference justification is delivered to the GTMS by a NM inference procedure.

Viewing the theory as standing for a set of conclusions:

$`p \in `t$

Viewing the theory as standing for a (possibly infinitary) sentence in some logical system, we can write this as an entailment. E.g., in the GCR architecture, viewing a theory as standing for a circumscription formula sentence (which is always finitary):

$$\models \ `t \supset `p$$

Or, in the GCR architecture, viewing the theory as standing for a CLD axiom set (when that is appropriate and possible[11]):

$$\models \ \mathcal{PDC}(`t) \ \supset \ `p$$

In the GCR architecture, this implication represents an entailment in (second-order) classical logic:

$$\mathcal{PDC}(`t) \ \models \ `p$$

; i.e.,

$$`t \ \approx_{CLD} \ `p$$

2) A monotonic inference (i.e., type-2) justification is the record of an ordinary, monotonic-logical inference: as in many previous applications of ATMSes.

$$`q1 \wedge \ldots \wedge `qn \ \supset \ `p$$

In the GNMR architecture, a monotonic inference is delivered to the GTMS by a monotonic or non-monotonic inference procedure.

In the GCR architecture, this implication represents an entailment in classical logic:

$$`q1 \wedge \ldots \wedge `qn \ \models \ `p$$

This is just equivalent to:

$$\models \ [(`q1 \wedge \ldots \wedge `qn) \ \supset \ `p]$$

3) A genealogical (i.e., type-3) justification is the record of a sub-theory relationship statement arising from a genealogical meta-inference. Abstractly:

tj is a sub-theory of ti

In the GNMR architecture, such a genealogical meta-inference is delivered by the meta-reasoning procedures that find decompositions and safeties. In section 7.7, we discuss how these genealogical meta-inferences and sub-theory relationship statements may be procedurally generated.

Viewing each theory as a set of conclusions, one theory is a sub-theory of another exactly when it is a subset:

$$g \ \Rightarrow \ h \qquad \Longleftrightarrow \qquad `h \subseteq `g$$

Viewing each theory as a sentence, e.g., a circumscription formula, one theory is a sub-theory of another exactly when the second logically implies the first:

$$g \ \Rightarrow \ h \qquad \Longleftrightarrow \qquad `g \models `h$$

Viewing each theory as a CLD axiom set (when this is appropriate and possible):

$$g \ \Rightarrow \ h \qquad \Longleftrightarrow \qquad \models \ \mathcal{PDC}(`g) \ \supset \ \mathcal{PDC}(`h)$$

; i.e.,

$$g \ \Rightarrow \ h \qquad \Longleftrightarrow \qquad \mathcal{PDC}(`g) \ \models \ \mathcal{PDC}(`h)$$

where $\mathcal{PDC}()$ is the circumscription operator for CLD that takes a CLD axiom set as input and

[11] Even in the application of the GNMR architecture to circumscription, the circumscription may be guarded. Or the circumscription may be more expressively powerful than PDC / CLD, e.g., [Grosof, 1992a]. Also, of course, even in PDC, and especially in PPC, one need not think in terms of CLD.

has a prioritized default circumscription as output.

Semantics of IN in terms of GNMR:
In terms of the GNMR architecture, the answer $IN(p,t)$ by the GTMS, where p is a proposition and t is a theory, should mean (more precisely, necessitates) that p is (known to be) a (sound) NM conclusion in the theory t. In the GCR architecture, the GTMS achieves this intended semantics: see Theorem 7.4 below. In the GCR architecture, viewing the theory t as a sentence, $IN(p,t)$ means that:

$$`t \supset `p$$

follows from the GCR meaning of the current set of justifications. Let J stand for this set, viewed as a conjunction of clauses. Then, we can write this as:

$$J \models `t \supset `p$$

or as:

$$\models J \wedge `t \supset `p$$

Remember that, in addition, the GTMS, like the ATMS, checks that t is not (known to be) inconsistent. (The account is similar for the other views of the meaning of a theory discussed above in connection with the GNMR meaning of a type-1 justification.)

Genealogical Justifications Arise From Decompositions and Safeties:
Genealogical justifications, i.e., sub-theory relationship statements, arise from meta-inferences about decompositions and safeties as follows. (See section 7.7 for more discussion of those meta-inferences.)

1. In every conjunctive decomposition: The global axiom set's NM theory logically implies each of its slices' NM theories.
   $$global \Rightarrow slice$$

2. In every serial decomposition: A global axiom set's NM theory logically implies each of its tiers (Definition 4.11).
   $$global \Rightarrow tier$$

3. After every globally monotonic update: the updated global axiom set's NM theory logically implies the previous global axiom set's NM theory.
   $$updated \Rightarrow previous$$

4. After every globally redundant (monotonic) update (Definition 4.17): The previous global axiom set's NM theory also logically implies the updated global axiom set's NM theory.
   $$previous \Rightarrow updated$$

In addition, the GTMS handles in an **IMPLICIT** fashion the following decomposition and safety relationships, using its genealogical justifications of the above four kinds.

5. In conjunctive decomposition: A global axiom set's NM theory logically implies the conjunctive combination of the NM theories for any subset of its slices. This is handled implicitly by

the GTMS in that any conclusion justified as IN by the conjunctive combination is justified as IN in the global.

6. In partially monotonic updates, where there are safeties arising from differential decomposition (Observation 4.6): A global axiom set's NM theory logically implies:

   (a) the (conjunctive combination of the) safe slices; and

   (b) the safe tiers.

   This is handled implicitly by the GTMS in that any conclusion justified as IN by the previous (conjunctive combination of) safe slices or safe tiers is justified as IN by the updated global. (Some examples of partial-monotonicity safety from differential decomposition include: higher-priority conclusions (section 5.12), disjoint sub-languages (section 6.2), and sympathetic updates (section 6.5).)

7. Hierarchical decomposition is handled appropriately. E.g., a global axiom set logically implies each slice of its slices, each slice of its tiers, each tier of its slices, etc.. This is handled implicitly by the GTMS in that the transitive closure of the genealogical justifications, i.e., the genealogy, is computed and used.

Example 7.5 in the next sub-section illustrates all of these behaviors, including the implicit ones.

What the GTMS Computes:
The GTMS does a limited degree of inference. Each time it receives a new justification, the GTMS performs propositional forward inference, using the current set of justifications (i.e., clauses). It compiles the information that it needs to be able to answer IN queries quickly, i.e., by lookup as in a database.

Let J stand for the set of current justifications, viewed as a conjunction of clauses. Let x be an ordinary / proposition node, OR an assumption / theory node. Let t be an assumption / theory node; t may NOT be an ordinary / proposition node.

Then the GTMS answers queries of the form $IN(x,t)$. These queries may be EITHER closed, i.e., instantiated to a particular x and t, OR open, i.e., with x and/or t variable.

The GTMS answers queries soundly and completely with respect to the current set of justifications. An instantiated query $IN(x,t)$ is answered affirmatively if and only if: BOTH

1. $J \models {}^{\backprime}t \supset {}^{\backprime}x$

   AND:

2. $J \not\models {}^{\backprime}t \supset \bot$

Here, the entailment is in (classical) propositional logic.

Open queries are handled accordingly.

Alternatively, the GTMS's inferences can be described in terms of the clauses it computes. The GTMS computes all of J's **prime implicates** [Reiter and de Kleer, 1987] that have the **form** '$t \supset$ 'x, where t is an assumption.

Consistency Checking within the GTMS:
In practical applications of GTMS to the GCR architecture, we expect the consistency condition (2.) above not to be much of an issue. It is pretty pointless to do inference in classical logic (in which circumscription is embedded) if the global theory is unsatisfiable. Typically, in applications, one assumes, or imposes conditions guaranteeing, satisfiability. E.g., each of the previous circumscriptive query-answering procedures discussed in section 7.4 impose conditions guaranteeing that the circumscription preserves the satisfiability of the base sentence.

Consistency checking within the GTMS can absorb a significant amount of computational effort. Yet, it is not complete with respect to the underlying semantics of the NM formalism.

Therefore:

1. One may not want the GTMS to do consistency checking at all. This can be accomplished by a straightforward **modification of the ATMS**: when maintaining / updating labels, just do not bother to check whether an environment (i.e., assumption / theory) is nogood. Note that, even with this modification, one can maintain a label and answer queries about the contradiction node $\perp$, treating it as just another ordinary node.

2. Whether or not one does want it to check consistency of all labels within the GTMS, it may be more useful to treat consistency as a separate query: $IN(p, \perp)$. It may not be appropriate simply to lump inconsistency with other kinds of "No" answers to $?IN(p,t)$ queries.

Observation 7.3 (Monotonicity of IN)
The set of IN conclusions in the GTMS is a logically monotonic function of the set of justifications, if EITHER:

1. None of the justifications have the contradiction node $\perp$ as consequent; OR

2. The GTMS is modified to omit consistency checking (see above),

GCR-Soundness of the GTMS's Inferences:
When applied to the GCR architecture, the GTMS's inferences are sound with respect to the underlying semantics of the circumscription NM formalism. Essentially, the GTMS is performing sound meta-reasoning about entailments / implications in higher-order logic. In application of the GTMS to CLD, the GTMS inferences are essentially a propositional fragment of second-order-logical inference. reasoning; e.g., that some conjunctions of circumscription formula sentences and base-language sentences imply other circumscription formula sentences and/or base-language sentences.

Theorem 7.4 (Soundness of GTMS for Circumscription)

Suppose that: with respect to its semantics in the GCR architecture, each justification in J is logically correct, i.e., sound, with respect to the circumscription (NM) formalism (e.g., CLD). (We described this semantics for justifications above, in connection with the three basic kinds of justifications.)

Then: GTMS query answering and labelling is logically correct, i.e., sound, with respect to the circumscription (NM) formalism (e.g., CLD). (We described this semantics above: the semantics for IN in terms of the GNMR architecture.)

Note that this result does not depend on the GTMS doing consistency checking (as discussed above).

Proof : Immediate from classical higher-order logic. □

Contrast to Other ATMS Applications:
The GTMS's application to the GNMR architecture differs in several important ways from the usual applications of the ATMS. The most significant are:

1. **Assumptions can be consequents** in justifications, and this is central. By contrast, in the standard ATMS ([de Kleer, 1986a], page 142), the notion of a justification was defined so as to permit the consequent to be an assumption, but "utilizing that feature was avoided", both in that paper and in the usual applications and analyses of ATMSes since. That is, a *de facto* **convention of usage** for ATMSes is that consequents are always ordinary nodes.

2. The spirit of use is **not abductive**.

3. The query interest is for **single-assumption environments** only. This point is a bit subtle. It may be that a NM conclusion follows from a conjunctive combination of two or more theories / assumptions. Yet, there is no need to keep that list of theories / assumptions as a label environment. The GNMR architecture is only really interested in when there is a single theory / assumption that is a super-theory of each of the members of that list.

In addition, two types of queries are more important than in the usual applications of ATMSes:

4. The **contents query (terminology)**: $IN(?p, t)$. I.e., to query what are all the ordinary nodes are IN that are IN in a particular environment. E.g., to query what are all the propositions (known to be) believed in the current global theory. That is, to query what are the "contents" of the current global theory.

5. The **genealogy query (terminology)**: $IN(?t, t)$. I.e., to query what are all the assumption nodes that are IN in a particular environment. E.g., what are all the (known) sub-theories of the current global theory.

Also, recall the discussion in section 7.3 of contrast with the usual ATMS applications.

7.5.5 Example GTMS Behavior in GNMR Application

In this sub-section, we illustrate some examples of what we expect the GTM bookkeeper to do, and which, in fact the GTMS does.

Example 7.5 (Behavior of the GTMS)

We present a running GNMR example: as a "blow-by-blow" account of the sequence of justifications received by the GTMS, and *some* of the IN conclusions the GTMS makes on the basis of those justifications. For the sake of conciseness, the theories and propositions appearing in the example are left abstract.

This example illustrates how the GTMS supports, in a significant and interesting manner, the ideas and distinguishing capabilities of the GNMR architecture for learning agents.

Conjunctive decomposition: In the learning agent, to begin with, the current global theory is $t1$. A meta-reasoning procedure finds a conjunctive decomposition for $t1$: $t2$ is a slice of $t1$. A NM inference procedure goes to work on the slice $t2$ and non-monotonically infers the conclusion p there. It delivers this NM inference as a justification to the GTMS:

$t2 \quad \Rightarrow \quad p$

Then the GTMS concludes:

$IN(p, t2)$

Next, the GTMS receives a genealogical justification based on the finding that $t2$ is a slice of the current global theory $t1$:

$t1 \quad \Rightarrow \quad t2$

Then the GTMS concludes that p, which was inferred by a NM inference procedure working on the axiom set / theory $t2$, is also a NM conclusion in $t1$:

$IN(p, t1)$

Conjunctive combination: Next, the GTMS receives another genealogical justification based on the finding that $t3$ is also a slice of $t1$:

$t1 \quad \Rightarrow \quad t3$

and a NM inference justification generated by a NM inference procedure working on the slice $t3$:

$t3 \quad \Rightarrow \quad q$

plus a monotonic inference justification generated by a monotonic inference procedure working on conjunctively combining the slices $t2$ and $t3$:

$p, q \quad \Rightarrow \quad r$

Then the GTMS concludes not only that:

$IN(q, t3)$

$IN(q, t1)$

but also r is sanctioned as a NM conclusion in the current global theory $t1$:

$IN(r, t1)$

Note that the GTMS *itself*, in effect, performs (part of) the conjunctive combination.

Global Monotonicity: Next, the global axiom set is updated. The new current global theory is $t4$. A meta-reasoning procedure finds that this axiom update is globally monotonic. Accordingly, it delivers a genealogical justification:

$t4 \quad \Rightarrow \quad t1$

The GTMS then concludes that all of the propositions that were sanctioned as conclusions in the previous global theory $t1$ are also sanctioned as conclusions in the current global theory $t4$:

$IN(p, t4)$

$IN(q, t4)$

$IN(r, t4)$

Global Redundancy: Next, a NM inference procedure working on the current global theory $t4$ finds the NM conclusion s there:

$t4 \quad \Rightarrow \quad s$

Then the GTMS concludes:

$IN(s, t4)$

Next, a meta-reasoning procedure finds that this latest axiom update is not only globally monotonic, but also is globally redundant. It delivers a genealogical justification:

$t1 \quad \Rightarrow \quad t4$

Then the GTMS concludes that s, which was inferred non-monotonically in the current global theory $t4$, is also sanctioned as a non-monotonic conclusion in the previous global theory $t1$:

$IN(s, t1)$

Next, a meta-reasoning procedure reports another slice $t5$ for the current global $t4$:

$t4 \quad \Rightarrow \quad t5$

and a NM inference procedure working on that slice reports a NM inference:

$t5 \quad \Rightarrow \quad e$

and a monotonic inference procedure working on conjunctively combining the current slices reports a monotonic inference:

$e, p \quad \Rightarrow \quad f$

Then the GTMS concludes that these propositions are also sanctioned as conclusions in the current global theory:

$IN(e, t4)$

$IN(f, t4)$

Safety via differential decomposition: Next, the global axiom set is updated again. The new current global theory is $t6$. A meta-reasoning procedure finds a decomposition for it: $t6$ has slices $t2$ and $t5$ (but not $t3$). This decomposition implies a partial monotonicity of updating, via differential decomposition. The previous global theory $t4$ had slices $t2$, $t3$, and $t5$. Thus, the conjunctive combination of slices $t2$ and $t5$ is safe under the update. Rather elegantly, this **safety via differential decomposition emerges from the GTMS operations**. How? The decomposition for the new current global $t6$ is reported:

$t6 \quad \Rightarrow \quad t2$

$$t6 \quad \Rightarrow \quad t5$$

The GTMS concludes that:

$$IN(p, t6)$$
$$IN(e, t6)$$
$$IN(f, t6)$$

That is, the GTMS concludes that each of the conclusions that it previously knew to be sanctioned in the conjunctive combination of $t2$ and $t5$ are sanctioned in the new current global theory $t6$. Note that the GTMS did not have to be told explicitly about partial monotonicity (safety) and differentiating the decomposition; it only had to be told the relevant decomposition of the updated theory. The GTMS then "carried over" the preserved conclusions automatically.

Hierarchical decomposition: Next, a meta-reasoning procedure finds a finer decomposition for the current slice $t2$: $t2$ itself is slice-able into $t7$ and $t8$. It reports:

$$t2 \quad \Rightarrow \quad t7$$
$$t2 \quad \Rightarrow \quad t8$$

The GTMS concludes that $t7$ and $t8$ are themselves sub-theories of the current global $t6$:

$$IN(t7, t6)$$
$$IN(t8, t6)$$

Note that the GTMS thus computes the **genealogy**, by using the transitivity of the genealogical justifications.

Next, a NM inference process working on the fine-grain slice $t7$ reports:

$$t7 \quad \Rightarrow \quad b1$$
$$t7 \quad \Rightarrow \quad b2$$

and another NM inference process **concurrently** working on the fine-grain slice $t8$ reports:

$$t8 \quad \Rightarrow \quad b3$$

and another monotonic inference process working concurrently reports:

$$b1, b3, e \quad \Rightarrow \quad b4$$

Then the GTMS concludes not only that the finest-grain slices' conclusions descend to the current slice $t2$:

$$IN(b1, t2)$$
$$IN(b2, t2)$$
$$IN(b3, t2)$$

but also that these conclusions descend to the current global $t6$:

$$IN(b1, t6)$$
$$IN(b2, t6)$$
$$IN(b3, t6)$$

which, further, leads the GTMS to conclude:

$$IN(b4, t6)$$

This illustrates how the GTMS applies the genealogy.

More synergy between safeties and decomposition: Next, the global axiom set is updated. The new current global theory is $t9$. A meta-reasoning procedure finds a conjuctive decomposition for $t9$: $t4$ and $t10$ are its slices. It reports:

$t9 \quad \Rightarrow \quad t4$

$t9 \quad \Rightarrow \quad t10$

The GTMS concludes that all of the conclusions already known to be sanctioned as conclusions in $t4$ are also sanctioned as conclusions in the current global $t9$. These include conclusions originally found by many different NM and monotonic inference processes while working on various different theories (/ axiom sets): $t2$, $t3$, and the conjunctive combination of $t2$ and $t3$ when the current global theory was $t1$; $t4$, $t5$, and the conjunctive combination of $t2$ and $t5$ when the current global theory was $t4$; and, $t7$ and $t8$ and the conjunctive combination of $t5$ and $t7$ and $t8$ when the current global theory was $t6$:

$IN(p, t9)$

$IN(q, t9)$

$IN(r, t9)$

$IN(s, t9)$

$IN(e, t9)$

$IN(f, t9)$

$IN(b1, t9)$

$IN(b2, t9)$

$IN(b3, t9)$

$IN(b4, t9)$

Note that here the GTMS **synergistically** combines descent on the basis of decomposition with descent on the basis of safety. E.g., it uses the genealogical justification arising from the meta-inference that $t4$ was the result of a globally monotonic update to $t3$.

Serial decomposition; mixed styles of decomposition: Next, a meta-reasoning procedure finds a finer, serial decomposition for current slice $t10$: $t1$ is (one of) its tier(s). It reports:

$t10 \quad \Rightarrow \quad t1$

The GTMS concludes that all of the propositions known to be sanctioned as conclusions in the $t1$ are also sanctioned as conclusions in the current global $t9$, as well as in the current slice $t10$:

$IN(p, t9)$

$IN(q, t9)$

$IN(r, t9)$

$IN(s, t9)$

$IN(b1, t9)$

$IN(b2, t9)$

$IN(b3, t9)$

Note this makes interesting use of the earlier global redundancy information discussed above. And it illustrates hierarchical descent through mixed modes of decomposition: i.e., through both conjunctive and serial modes.

7.5.6 Algorithms and Complexity Analysis

In this sub-section, we discuss how the GTMS can be implemented, and what the computational cost of its operations are.

Remember, the GTMS is a strict special case of the standard ATMS. Therefore, **the GTMS can be implemented directly using any of the previous implementations / algorithms for the standard ATMS, e.g., de Kleer's.**

However, one would like to take advantage of the special case properties of the GTMS. Selman and Levesque [1990] showed that the ATMS task, i.e., ATMS maintenance / updating, is NP-hard. Intuitively, this is because, in the worst case, an exponential number of different assumption sets (subsets of the overall set of assumptions) need to be checked, in effect, for whether they (more precisely, their conjunction) implies each node proposition.

By contrast, in the GTMS, only singleton assumption sets need to be checked. The number of these is worst-case linear in the number of nodes.

The GTMS task, i.e., GTMS maintenance / updating, is computationally tractable.

Theorem 7.6 (Computational Tractability of GTMS)

For a given set J of justifications, all GTMS labels (equivalently, the set of all ground $IN(x,t)$ pairs) can be computed in time:

$O(a \cdot n \cdot j)$

where a is the number of assumption nodes appearing in J, n is the number of nodes (ordinary and assumption) appearing in J, and j is the size of J. By the size of J, we mean the number of letters appearing in the clauses J: i.e., the number of consequents plus the number of antecedents.

Note that since $a \leq n \leq j$, the above is upper-bounded by:

$O(j^3)$

Proof : Our proof is constructive. It includes the sketch of a **brute-force algorithm**.

Let x be any node, and let t be an assumption node. We view J as a conjunction of clauses.

Step 1 of the algorithm: For each pair (x, t), compute whether:

$J \models \text{`}t \supset \text{`}x$.

We next show that the question:

$J \models \text{`}t \supset \text{`}x$ can be computed in time $O(j)$.

This question is equivalent to the question:

$J \wedge \text{`}t \models \text{`}x$

Since each clause on the left-hand-side is propositional Horn with non-empty head, this question can be computed in time linear in the size of the left-hand-side. The fact that such query-answering is linear-time is well-known in the Datalog literature. For example, see the results and **algorithms** of Dowling and Gallier [1984], and Scutellà's [1990] follow-on correction to their work. The clauses $J \wedge t$ can be viewed as a propositional Datalog program. [12]

The number of such (x, t) pairs is $n \cdot a$.

Thus all of the GTMS labels *before (/ without) consistency checking* can be computed in time $O(a \cdot n \cdot j)$.

Step 2 of the algorithm is to prune all of the inconsistent labels.

The contradiction node $\perp$ is an ordinary node. Thus step 1 computes its label. To check this label against each label resulting from step 1, and prune, can be computed in time:

$O(a^2 \cdot n)$

since there are n labels, and each label has length $O(a)$.

Remember, each label corresponds to a set of single-assumption environments, and, therefore, to a subset of the overall set of assumptions.

This comparison can be implemented with bit vectors, e.g., schematically as:
$(AND(x - label, NOT(\perp{-}label))$
where each label has one bit position per assumption.

Finally, since $a \leq n \leq j$, the overall cost of steps 1 and 2 is $O(a \cdot n \cdot j)$.

□

The above complexity analysis, and algorithm sketch, was for the **batch** case of GTMS updating, beginning from a state with no previous justifications. I.e., it was for a batch of justifications, "starting from scratch". The more typical GTMS operation, however, is **updating** with a new justification, and revising the (non-trivial) previous GTMS labels accordingly. Thus the above complexity upper-bounds the computational cost of this updating operation. And the above algorithm is a **brute-force updating algorithm**, as well as a brute-force batch algorithm. For each justification update, it throws away all previous work and starts from scratch.

The other GTMS operation is query-answering. This is by look-up, as in a database. It can be computed in time

$O(log(j))$

per instantiated IN pair.

A less brute-force approach to algorithm design is to **adapt a standard ATMS updating algorithm** (e.g., de Kleer's). Elsewhere, we show how to do this relatively efficiently.

There are some other considerations in implementing a GTMS, relative to the usual ATMS implementations, that arise from the differences in the GNMR application from the usual ATMS applications. One may want to add extra data structures to facilitate contents queries. And one may want to handle consistency checking in a different way. (See discussion of "Contrast to Other ATMS Applications" and "Consistency Checking within the GTMS" toward the end of sub-section 7.5.4.)

Elsewhere, we discuss GTMS algorithms and implementation in more detail.

[12] Datalog is a standard concept in the logic programming literature. For example, see [Ullman, 1988], especially pages 100-106, for a description.

7.5.7 Previous Applications of Truth Maintenance to NMR

There exist two previous styles[13] of application of truth maintenance systems (TMSes) to support non-monotonic reasoning. In the first style, the (inference purely within the) TMS itself is logically non-monotonic; that is, the set of currently IN nodes is a logically non-monotonic function of the current set of asserted justifications. This first style was pioneered by Doyle [1979]. In his TMS, within a justification, the consequent being IN is implied by an overall antecedent condition which may include the OUT-ness of some nodes as well as the IN-ness of other nodes. (OUT means: not IN.)

In the second style, an Assumption-based TMS (which is itself logically monotonic) is used to support abduction. Abduction in turn supports default reasoning. This style was pioneered by de Kleer [1986b]. [Selman and Levesque, 1990] describe this style at a more abstract level than de Kleer did, and give a complexity analysis. Other related work sharing this style of approach includes: [Ginsberg, 1989] [Junker, 1991] and [Raiman and de Kleer, 1992].

Our Genealogical approach to applying truth maintenance to NMR (i.e., the GTMS and its role in the GNMR architecture) is very different from both of these previous styles.

7.6 Mechanizing Specification in CLD; Reformulating Abnormalities

In this section, we sketch how to mechanize CLD as a specification tool for PDC or PPC. **Recall the overview of this section in section 7.3.**

This section discusses kinds of reformulation at the level of axioms, i.e., of the specification.

The results in next section build on the results in this section. Mechanizing specification in CLD is a kind of pre-processing or sub-procedure that is often used by our decomposition and safety analysis procedures.

Note on complexity analysis: For the sake of brevity, we **ignore logarithmic factors**, for the most part, in this and the next section.

7.6.1 A CLD Interpreter

In this sub-section, we sketch a **CLD interpreter (terminology)** procedure that takes, as input, a sequence of CLD axiom updates, then converts the current CLD axiom set into a PDC. We also sketch how to convert PPC or PDC to CLD, and PPC to PDC: these are straightforward. We show that all this can be done in time that is either **linear** or **quadratic** in the size of the CLD axiom sets involved.

We begin with some preliminary definitions.

For sake of simplicity and concreteness of discussion: Here and in the next section, we **define data-structure representations** of PPC, PDC, and CLD as follows. We are not concerned here

[13] in the NMR literature, that we are aware of

with the details of efficiency; rather, we wish to mirror the mathematical syntax in a straightforward fashion.

In this and the next section, often omit saying "data-structure representation of" when it is is clear in context that we are talking about the data-structure, as opposed to the mathematical, sense of representation.

We take the base language of the circumscription to be first-order. For concreteness, we adopt standard first-order predicate calculus as the syntax of this language, and represent this syntax in the data structure as (something basically like) a character string. Each base, default, and fixture formula is in the base language. E.g., each might be a conjunctive list of clauses. (More generally, in the case of serial circumscriptions, the base language may be second-order predicate-calculus, however. Also, uniqueness of names and domain closure can be handled in a more directly computationally efficient fashion than the above standard syntax.)

We define (the data-structure representation of) a PPC as: a three-field record. The first field is a (conjunctive) list of base sentences, e.g., a list of clauses. The second field is a list of minimized predicate symbols. The third field is the prioritization partial order (binary) relation: an asymmetric 2-dimensional matrix. The indices of this p.o. refer to those in the list in the second field. The fourth field is a list of fixed predicate symbols. (This can easily be generalized to permit functions to vary: make it instead a list of fixed function and predicate symbols.)

We define (the data-structure representation of) a PDC similarly to (that of) a PPC. There are only two differences. One is that the second field is, instead, a list of maximized default formulas, in first-order predicate-calculus. Each member of the list may itself be a conjunctive list of clauses, for example. The other difference is that the fourth field is, instead, a list of fixed formulas.

A PPC is a special case of a PDC: just view each minimized predicate as negated literal formula, and each fixed predicate as an atomic formula.

We also define (the data-structure representation of) a CLD axiom set as a four-field record. Each field corresponds to a type of axiom. The first field is a list of base axioms: each is a formula in the base language, just as in the PPC and PDC representations. The second field is a list of default axioms. Each member of the list has two sub-fields. The first sub-field is its label: some symbol (remember, this symbol is meta-linguistic relative to the base language). The second sub-field is its formula part: a maximized default formula. The third field is a list of prioritization axioms. Each prioritization axiom is an ordered pair of default axiom labels. The fourth field is a list of formula axioms: each is a formula in the base language.

There are two main differences in a CLD axiom set compared to a PDC. The most significant is that **the prioritization partial order relation is only implicit**, rather than explicit. The prioritization p.o. is the **transitive closure** of the relation specified by the union of the pairs that correspond to the prioritization axioms. The second difference is that the default axioms have a label sub-field.

PPC is a special case of PDC. Thus procedurally converting PPC to PDC takes zero computation.

Procedurally **converting PPC / PDC to CLD** takes time that worst-case linear in the size of the input. The **algorithm sketch** is: Each default formula is given a label, and each pair in the prioritization partial order relation is made into a prioritization axiom. Note that the size of the input prioritization partial order is, worst-case, *quadratic* in the number d of input default formulas. Thus, worst-case, the size of the input is quadratic in the size of: the union of the base, default, and fixture formulas.

More generally, there are, typically, many CLD axiom sets that are equivalent to (i.e., that specify) a given PDC. This non-determinism arises from the fact that many different collections of prioritization axioms may have the same transitive closure. We ignore this point in the rest of our discussion, for the most part.

Earlier (especially in sections 1.3, 1.6, and 3.2), we discussed the desirability of using CLD as a tool for specification. CLD enables one to *incrementally* specify a PDC. **Recall point 2. in sub-section 7.1.2.**

In the GCR architecture, it is thus useful that one of the available meta-reasoning procedures be a CLD interpreter.

Next, we give an **algorithm sketch and complexity analysis for the CLD interpreter**.

Procedurally converting a **batch** CLD axiom set to PDC is a bit more complicated than vice versa, due to the need to perform transitive closure on the prioritization axioms in order to construct the prioritization partial order relation.

We **observe** that, using, for example, the results and algorithms of [Kohavi and Shoham, 1992], this transitive closure can be performed in time:

$O(s \cdot d)$

where d is the number of defaults; and s is the number of prioritization axioms. Kohavi and Shoham's method uses "linear rules" in Datalog. Thus the overall conversion cost is:

$O(n + (s \cdot d))$

where n is the size of the union of the base, default, and fixture formulas. This is uppper-bounded by $O(m + (s \cdot d)) \leq O(m \cdot s) \leq O(m^2)$

where m is the total size of the input CLD axiom set (representation).

Interestingly, Kohavi and Shoham's method can also be used to efficiently **incrementally update** a transitively closed relation: the above cost can be amortized (i.e., averaged) over the updates. Thus updating with a new prioritization axiom costs **amortized** time:

$O(d)$

Incrementally updating with a new base, default, or fixture axiom is very simple.

Let u be the size of a CLD axiom update. To summarize: the time cost of **processing a CLD axiom update** is:

$O(u)$

for base, default, and fixture axioms; and the cost is:

$O(u \cdot d)$

for prioritization axioms.

7.6.2 Default Case Via Predicate Case

In this sub-section, we sketch how, procedurally, to expressively generalize any available procedure for inference (or belief revision) in PPC so as to apply it to PDC. The key step in our method is to introduce abnormality predicates. **Recall point 1. in sub-section 7.1.2.**

Our approach exploits our earlier results about abnormality theories in sub-sections 3.3.2 and 3.5.4. The **algorithm sketch and complexity analysis** is as follows. We stick an input transformer and an output transformer on the PPC inference (or belief revision) procedure. The input transformer converts the input PDC to a PPC, which is then input to the PPC inference (or belief revision) procedure The output transformer converts conclusions output by the PPC procedure, into conclusions for the PDC. To get the input transformer: Corollary 3.46 (and Theorem 3.43 if needed; see below) is procedurally directly implemented, in the direction from the PDC to the adaptive-form abnormality PPC. This is straightforward. The time cost is at most **linear** in the size of the input PDC, or, alternatively, in the size of the corresponding CLD axiom set. The result is a conservatively extended circumscription. The output transformer performs the second-order existential projection: it simply filters the output of the PPC so as to prune any conclusions that mention any of the newly-introduced predicates (abnormalities and fixed parameters). For many practical purposes, this output transformer can be dispensed with. The output transformer cost is at most linear in the size of the output.

Using this **DP-converter (terminology)** ("DP" is mnemonic for "Default case via Predicate case") approach, we can thus expressively generalize all of the previous inference procedures available for PPC that we discussed in section 7.4. This expressive generalization involves trivial overhead (see above). In particular, we arrive at **new inference procedures for prioritized DEFAULT circumscription**, for two expressive sub-classes:

1. **parallel priority, ground, uniqueness of names**: by applying the DP-converter to Przymusinski's [1989] or Inoue and Helft's [1990] algorithms.

2. **layered priority, domain closure, uniqueness of names, no functions of non-zero arity**: by applying the DP-converter to Baker and Ginsberg's [1989] algorithm. (This uses Theorem 3.43.)

Thus the DP-converter is another useful procedure to include in the GNMR architecture.

7.7 Recognizing and Performing Decompositions and Safeties

In this section, we sketch how to implement most of our major results on decompositions, safeties, reformulations, and inference procedures from chapters 2 through 6. We observe that the computational cost involved, both to recognize the relevant conditions / special cases and to perform the decompositions etc., is polynomial in the size of the global CLD axiom set.

7.7.1 Introduction and Overview

Tractable Overhead in GNMR Architecture:
A prime motivation behind the GNMR architecture is to pick off special cases and thus *sometimes* to beat the general intractability of the NM aspect of NMR (sub-sections 4.2.2 and 4.3.1). In doing so, it is desirable that the GNMR architecture itself impose only tractable overhead beyond the basic NM inference and belief revision procedures that it calls. The key elements of the GNMR architecture besides these basic procedures are the procedures that perform:

1. meta-reasoning about decompositions and safeties

2. specification meta-reasoning, e.g., converting between CLD, PDC, and PPC

3. genealogical truth maintenance: e.g., the GTMS

In addition, there are the procedures that perform the overall control of the architecture.

In sub-section 7.5.6, we showed this tractability for the GTMS.

In section 7.6, we showed this tractability for circumscriptive specification-oriented meta-reasoning: in particular, for the CLD interpreter and the DP-converter.

In this section, we sketch this tractability for GCR meta-reasoning that exploits our other results about decompositions, safeties, reformulations, and inference procedures. We sketch algorithms to greater and lesser extents. We do not always give tightest complexity bounds. For the sake of brevity we often **ignore logarithmic factors in the complexity analysis**.

Syntax and Easy Meta-Reasoning:
Part of our strategy of attack on the difficulty of NMR is find structure and special cases that can be easily recognized on the basis of the syntax of the axiom set. (E.g., recall the summary of detailed strategy in section 4.6.)

Such **syntactic recognizability** makes for **easy meta-reasoning** about decompositions etc..

Most of our major results on decompositions and safeties (which involve, also, reformulations and new inference procedures) in chapters 2 through 6 fulfill this goal: their conditions are defined in terms of syntactic attributes of the CLD axiom set. In this section, we observe that most of those results, therefore, can be **exploited procedurally** in time **polynomial in the size of the CLD axiom set**. Thus these results can be implemented as meta-reasoning procedures and used in the GCR architecture while imposing tractable overhead. Thus all of the key elements in the GCR architecture can be implemented with only tractable overhead, for expressively rich, interesting cases.

Outline of Section:
In the following sub-sections, we discuss how to implement and to exploit practically, in the GCR architecture, various of our results from chapters 2 through 6: these we present in almost the same sequence as they appeared there, i.e., "chronologically".

For the sake of simplicity of discussion, in the remainder of this section, unless we mention explicitly otherwise, we will **assume** that the circumscription is a PDC, e.g., a PPC, and that this circumscription is **specified in CLD**. (See section 7.6 on how the relationship between CLD and PDC and PPC may be implemented.)

Also: **Recall points 3. through 6. in sub-section 7.1.2.** Our new special-case forward inference procedures can be added to the library of circumscriptive inference procedures available in the GCR architecture.

7.7.2 Representation of Meta-Reasoning about Decompositions and Safeties

In this sub-section, we discuss some of the issues in implementing GCR meta-reasoning that are common to various cases of decompositions and safeties.

Meta-Reasoning Declaratively Viewed:
The phrase "meta-reasoning" has a perhaps nebulous ring to it. Next, we explain more concretely what kind of meta-reasoning we have in mind: in the context of implementing our decomposition and safety results for the GCR architecture.

Reasoning about decompositions and safeties (and reformulations) is "meta-" relative to particular circumscriptions / axiom sets / theories. This is why we call it "meta-reasoning". This meta-reasoning can be viewed declaratively: as inference in a logical meta-language. This is why we also call it "meta-inference".

A conjunctive decomposition is an equivalence between a circumscription and a conjunction of circumscriptions. In section 7.6, we described an implementational representation of a circumscription. A decomposition can be described meta-linguistically to CLD: by using the names of the circumscriptions or their corresponding axiom sets. In describing the GTMS, we used such a representation: a GTMS-theory / ATMS-assumption was exactly such a name. Computationally, of course, such a name can be implemented with a pointer.

Next, we sketch one simple such meta-language for decompositions and safeties that suffices for our discussion in this section. **Notation:** We will call this meta-language of decompositions and safeties: $\mathcal{MD}$. The meta-language describes properties of axiom sets / circumscriptions.

$\mathcal{MD}$ is a first-order-logical language. Thus the ("meta-")inferences within it are logically monotonic. Standard first-order inference methods can be applied to reasoning within it. Next, we give some of its ontology.

The conjunctive decomposition of an axiom set $\mathcal{A}_1$ into two axiom sets $\mathcal{A}_2$ and $\mathcal{A}_3$ can be represented as:

$$CONJ\_DECOMP(t1, \{t2, t3\})$$

Here, for $i = 1, \ldots, 3$, ti represents $\mathcal{A}_i$. Alternatively, each ti could represent, i.e, denote, the corresponding circumscription: $\mathcal{PDC}(\mathcal{A}_i)$, or the corresponding circumscriptive theory.

$CONJ\_DECOMP(x, \{y, \ldots, z\})$ is a predicate in the meta-language $\mathcal{MD}$. It has variable arity. We write it as a binary relation whose second argument is a (finite) set.

Similarly, the serial decomposition of an axiom set $\mathcal{A}_4$ into a sequence of two axiom sets $\mathcal{A}_5$ and $\mathcal{A}_6$ can be represented as:

$SERIAL\_DECOMP(t4, \langle t5, t6 \rangle)$

$SERIAL\_DECOMP(x, \langle y, \ldots, z \rangle)$ is also a predicate in $\mathcal{MD}$, with variable arity. We write it as a binary relation whose second argument is a (finite) tuple.

The global monotonicity of an update $\mathcal{A}_8$ relative to a previous axiom set $\mathcal{A}_7$ that results in a new axiom set $\mathcal{A}_9$ can be represented as:

$GLOBAL\_MON(t7, t8, t9)$

In similar fashion, as many attributes of axiom sets / circumscriptions / theories can be described as are desired for the purposes of meta-reasoning about decompositions and safeties. Some further examples of such properties are:

- One axiom set is a slice of another: e.g., $SLICE(t1, t2)$ above.
- One axiom set is a tier of another: e.g., $TIER(t4, t5)$ above.
- One axiom set's theory is a sub-theory of another's: e.g., $SUB\_THEORY(t1, t3)$, $SUB\_THEORY(t4, t5)$, $SUB\_THEORY(t9, t7)$ above.

In such a fashion, many other properties and relationships can be represented: one axiom set includes another; one axiom set is equal to the union of a set of other axiom sets; an axiom set contains no prioritization axioms; an axiom set's prioritization partial order is layered; that one circumscription is the conservative extension of another; that one circumscription is the result of guard reduction via introduced guard predicates, from another; clausality of the base and default and fixture formulas; their universality; the quasi-propositionality of a circumscription; the inclusion of the uniqueness of names in the base of an axiom set; a circumscription is sympathetic-solitary; and so on. A formula E in a circumscription's base language can also be represented as an object in $\mathcal{MD}$: i.e., the name of E.

One can, in addition, represent characterizing axioms for this ontology in the logical meta-language. E.g., that:

$SUB\_THEORY(,)$ is transitive;

$CONJ\_DECOMP(x, \{y, \ldots, z\})$ implies $SLICE(x, y)$;

$SLICE(x, y)$ implies $SUB\_THEORY(x, y)$;

parallel implies layered, and so on.

The meta-reasoning procedures about decompositions and safeties need not be implemented via first-order-logical theorem-proving methods; however, they can use such methods for part of their activities. Thus, they might share some common knowledge bases of information representable in a meta-language like $\mathcal{MD}$.

Utilizing the GTMS:

The above kinds of meta-inferences about decomposition and safeties immediately yield **genealogical justifications** that can be asserted to the GTMS. For example, from $SLICE(t1, t2)$, an appropriate genealogical justification is:

$t1 \quad \Rightarrow \quad t2$

Similarly, from $GLOBAL\_MON(t7, t8, t9)$, an appropriate genealogical justification is:

$t9 \quad \Rightarrow \quad t7$

Likewise, for $TIER(,)$.

$SUB\_THEORY(,)$ above plays a role similar to a genealogical justification or a genealogical IN pair in the GTMS. The GTMS takes care of its transitivity.

As we discussed in section 7.5, **partial-monotonicity safety from differential decomposition** is an emergent property, in effect, within the GTMS; all the GTMS need be told is the pair of decompositions. *Finding* a pair of decompositions before and after an update that make for such a useful differential, however, is the role of meta-reasoning procedures about safety.

More generally, the meta-reasoning about decompositions and safeties can use the GTMS as a resource. E.g., to query whether one circumscription is known to be a sub-theory of another.

Guide to Reader: CLAIM STATUS in Rest of Section: The rest of this section takes the form of (original-by-us) **OBSERVATIONS**, except where explicitly mentioned otherwise.

7.7.3 Monotonicity of Prioritization

By Theorem 2.58, updating with any prioritization axiom is globally monotonic. It is very simple to recognize this special case of an update. The fact that the update axiom is of the prioritization type is syntactic within CLD. The "performance" of the global monotonicity can also be performed very easily: by asserting a genealogical justification to the GTMS. (And, perhaps, asserting a $GLOBAL\_MON(,,)$ fact to some other meta-reasoning knowledge base.) Overall, this recognition and performance can be computed in constant time.

7.7.4 Formulas Purely in Fixed Symbols

By Theorem 3.58, any formula E that is fixed relative to a circumscription is globally monotonic as a base update or as default update with any priority. "Fixed relative to" includes both explicit fixing and indirect fixing. In general, fixture "relative to" may not be easily detectable: it is not a surface-syntactic property. However, for an important broad case this fixture is, indeed, surface-syntactic: when E is a formula purely in explicitly fixed symbols (fixed predicates and functions).

Procedurally, this case can be recognized in time:

$O(u \cdot f)$

where u is the length of the update formula, and f is the number of fixed symbols. Note that f is bounded above by the overall size n of the CLD axiom set.

Performance of the global monotonicity is simple: it requires only constant time: see discussion in the last sub-section.

7.7.5 Canonical Decomposition Along Prioritization

Decomposing does not necessitate having to compute all the conclusions in the slices or tiers, and then to combine them to form the global theory. Decomposing an axiom set / circumscription just means reformulating it into a tuple of other axiom sets / circumscriptions.

Canonical decomposition along prioritization (chapter 5), one default (/ minimized predicate) per slice or phase, can always be performed. Recognizing its applicability is thus trivial, including procedurally. Performing such a decomposition procedurally is accomplished by computing a data-structure representation of each of the slices or phases: each is just a CLD axiom set (or, if you like, a PDC). Recall that in sub-section 7.6.1, we described such data-structure representations for CLD axiom sets. The computational complexity of the canonical decomposition task is straightforwardly extracted from the expressive analysis in section 5.11: it is polynomial in the size of the starting global CLD axiom set.

Canonical decomposition along prioritization can also be done **hierarchically**, as well as one-by-one. Decomposing into one slice or phase per default **instance** (Theorem 5.29) is applicable when the overall circumscription contains domain closure. Domain closure is typically implemented in practice as a special kind of base axiom; thus its presence is usually easy to detect, e.g., in constant time.

Thus, typically, *recognizing* when to apply instance-grain canonical decomposition along prioritization can be computed tractably. *Performing* the decomposition is also tractable, just as for one-by-one above: there is one slice per instance. By domain closure the number of these instances is polynomial in the size n of CLD axiom set, since the size of the domain is bounded above by n. (We **assume** that the maximum arity of any default is bounded.)

More generally, the applicability of hierarchical canonical decomposition along prioritization depends on **detecting suitable (i.e., interesting) decompositional structure within the global prioritization partial order.** Composition of prioritization partial orders can be performed in polynomial time. However, we **observe** that the set of all decompositions of a prioritization partial order may be combinatoric: exponential in the size of the partial order order, and thus, in the size of the global CLD axiom set. E.g., in a total prioritization order, one can view the overall order as a discretized line; every segmentation of that line then corresponds to a (non-trivial) decomposition of the overall prioritization.

Nevertheless, many interesting cases can be recognized in polynomial time: e.g., totality, totally-prioritized modules, parallel composition of columns. That is, one can procedurally analyze the prioritization partial order and, in general, extract some of its interesting decompositions. Moreover, for restricted classes, detection of all decompositions is possible tractably.

Thus *recognizing* all possible ways to apply hierarchical canonical decomposition along prioritization is, in the general case, intractable. However, in practice, one does not want all possible decompositions, and many interesting special cases can be recognized tractably. Thus, we do not see this intractability as too worrisome. The harder problem, which is specific to the domain of application of the learning agent, is to operationalize "interesting" above.

Performing the hierarchical decomposition is, again, tractable, for the same reasons as in the one-by-one case.

Implications of One-by-One Canonical Decomposition:

Recall points 4. and 5. in sub-section 7.1.2.

Guard Reduction:
A further technical note on guard reduction. The second-order-logical existential projection $\exists GPT.$ of the newly-introduced guard predicates GPT is simple to accomplish procedurally: just filter the output of NM inference in the slice to prune away all conclusion sentences in which any of the newly-introduced guard predicates GPT appear.

7.7.6 Safety of Higher-Priority Conclusions

The safety of higher-priority conclusions (section 5.12) can be performed procedurally as follows. Canonically decompose conjunctively along prioritization, e.g., one-by-one, before and after the update. After each decomposition, assert the corresponding genealogical justifications to the GTMS: one per slice. Partial-monotonicity safety via differential decomposition then emerges as a property from the GTMS's labelling inferences, as discussed in section 7.5. Performing both the canonical decompositions and exploiting the GTMS are tractable tasks, as we discussed in the last sub-section and in sub-section 7.5.6, respectively. Recognizing procedurally the applicability of the safety of higher-priority conclusions is very simple: the update need only be a default update (default axiom plus prioritization axioms just about that default axiom's label). This requires time at most quadratic in the size of the update. To recognize the more general condition of a module update is not much more difficult, and is also tractable: polynomial in the size of the CLD axiom set.

There may be quite a few slices in each of the canonical conjunctive decompositions above (i.e., before and after the update). Thus, it may save computational effort if the one-by-one slices can be aggregated. One method of aggregation is to perform the decompositions hierarchically, e.g., by prioritization as we discussed in the last sub-section. Another method of aggregation is to represent a conjunction of (a subset of the overall set of) slices as a single GTMS theory.

A final note: the special case of a lowest-priority default or module update is especially easy computationally to recognize and to perform. To perform it, a single genealogical justification is asserted to the GTMS, arising from the global monotonicity.

7.7.7 Totally-Prioritized Modules

Serial decomposition by totally-prioritized modules (sub-section 5.13.1) is a special case of hierarchical canonical decomposition along prioritization (see sub-section 7.7.5). As we discussed there, it can be *performed* procedurally tractably.

However, we **observe** that the set of all possible ways of applying decomposition by totally-prioritized modules may be combinatoric (exponential): there may be a combinatoric number of ways to decompose the global prioritization p.o. into totally-prioritized modules. E.g., if the global prioritization is total: recall our earlier observation in sub-section 7.7.5.

However, all is not lost. For practical purposes, procedural *recognition* of totally-prioritized modules can be computed tractably. Practical usage of the serial decomposition result requires only being able to find all of the ways that a prioritization partial order can be decomposed into two modules such that one is strictly higher-priority than the other. This 2-ary case can then be applied recursively.

The following algorithm handles the 2-ary case. First, topologically sort the global prioritization partial order, by descending priority sequence. Topological sorting (Definition 5.31) is non-deterministic, but this does not matter: just make some non-deterministic choice. The result of sorting is a sequence s of indices (default labels). Then, for each prefix (i.e., initial sub-sequence) a of s, test whether: every index in that prefix a has higher-priority than all of the indices in the suffix (i.e., remainder) b of the sequence s. Each prefix that passes the test corresponds to a 2-ary higher-priority module; more precisely, the set of indices in that prefix are the indices in that module. The suffix corresponding to that prefix corresponds, in the same fashion, to the lower-priority module in that 2-ary decomposition of the prioritization partial order. The prioritization internal to these higher- and lower- priority modules can then be extracted straightforwardly from the global prioritization partial order: by projection / selection. All this can be done in time polynomial in the size of the global CLD axiom set.

Thus, for practical purposes, *recognition*, as well as performance, of serial decomposition by totally-prioritized modules is tractable.

7.7.8 Totally-Prioritized Propositional Defaults

Recall that, in sub-section 5.13.2, we showed that a special, more nicely behaved form of serial decomposition applies to circumscriptions with total prioritization and propositional defaults. This led to a **new, special-case forward inference procedure**: applicable either exhaustively (with respect to the NM aspect) or selectively. This special case forward inference procedure can thus be **added to the library** of circumscriptive inference procedures available in the GCR architecture.

Recognizing this special case is straightforward. Totality of the prioritization p.o. can be tested directly from the data structure representation of that p.o.., in time linear in the size of the prioritization. Propositionality of the defaults can be recognized from surface syntax, in time linear in the size of the global CLD axiom set. Lack of fixtures can be detected in like fashion.

Performing the serial decomposition is linear in the size of the defaults, and thus in the size of the global CLD axiom set.

The inference procedure for this special case calls, as a sub-procedure, a satisfiability test on a propositional default. This satisfiability test may be intractable: satisfiability testing is NP-complete for propositional languages. The overall inference procedure (for totally-prioritized propositional defaults) calls this satisfiability test once per default. The number of such calls is at most linear in the size of the CLD axiom set. Thus the (NM) inference procedure for this special case is also NP-complete. This complexity is the same as difficulty of *monotonic* inference, in the general case, for propositional languages. The complexity in this special case (with propositional base, as well as propositional defaults) is thus better than the general case of NM inference in circumscription over propositional languages. In effect, the NM aspect is no harder than the monotonic aspect of reasoning, unlike in the general situation of expressive NMR (recall the discussion of complexity in sub-sections 4.2.2 and 4.3.1.).

7.7.9 Disjoint Sub-Languages

Section 6.2 discussed the clean conjunctive decomposition available for the special case of disjoint sub-languages. This special case is defined in terms of surface syntax except for the condition on prioritization which involves some subtleties (see below).

To find the finest-grain syntactic partition of the base and default axioms (where only overlap of fixed predicates and functions is permitted) is straightforward to proceduralize. This analysis costs time polynomial in the size of the global CLD axiom set. (Here, we are **assuming** that the fixing of symbols is explicit, as is typical in applications.) By "finest-grain" here, we mean the partition with the finest, i.e., smallest, elements, and the largest number of elements. By merging elements of a k-element partition, of course, coarser-grain partitions can be generated.

The sufficient prioritization condition in Theorem 6.1 has three cases. The first case ((0.1) in the Theorem) is layered-ness. This can be tested for the global prioritization p.o. in time polynomial in the size of the global CLD axiom set: by testing directly for condition 3. of Observation 2.50 in the global prioritization p.o. relation, for example.

The second case of the sufficient prioritization condition ((0.2) in the Theorem) is that the global prioritization p.o. is modular by partition. For any given partition, to test this modularity can be performed in time polynomial in the size of the global CLD axiom set, as follows. Consider the partition of just the defaults. This partition has elements: each is a subset of the defaults, and thus corresponds to a subset of the default labels, i.e., of the indices in the global prioritization p.o.. Consider each index i in the global prioritization partial order relation. i belongs to exactly one partition element p. Of the global set of indices, some are *internal* to (i.e., members of) p, and some are *external* to (i.e., not members of) p. Test, for each i, whether i has the same relative priority towards each external index j, as does every other internal index k. I.e., for every internal k and external j, test whether i and k have the same relative priority towards j.

(The third case of sufficient prioritization is discussed below.)

In our experience with examples and applications, we have found that the above testing, which is tractable, suffices for practical purposes. If the conditions in the Theorem are met at all, they are met by the above testing: i.e., find the finest-grain partition of the base and default formulas, and test for layered-ness or modularity of the prioritization. (Note that modularity is relative to the partition). This testing corresponds to a broad special case of the Theorem.

Some perspective on the above practicality is that syntactically disjoint groups of base and default axioms tend to be associated with disjoint conceptual topics. Typically, the prioritization axioms specified are also grouped by these topics, in such a way as to be (finest-grain) modular (when not layered).

Having said that, we remark that, however, there are some additional subtleties, and worst-case, things can get nasty computationally in attempting to test for the conditions of the Theorem in their full generality. In general, the prioritization conditions (0.2) (modular) and (0.3) (upper bound is modular or layered) in the Theorem might be met for some coarser-grain partition of the base and defaults even if they are not met for the finest-grain. Worst-case, there are a combinatoric (exponential in the size of the global CLD axiom set) number of such coarser-grain partitions.

We remark, further, that prioritization condition (0.3) (upper bound is modular or layered) in the Theorem is, as yet, primarily of mathematical interest: we have not found any natural examples that satisfy it that do not satisfy modularity or layeredness.

To summarize the above: recognizing the special case where the disjoint sub-languages decomposition Theorem applies is tractable, for practical purposes: polynomial in the size of the global CLD axiom set.

Procedurally *performing* the conjunctive decomposition is then straightforward, and can be computed in time polynomial in the size of the global CLD axiom set. Just as in the other cases of decomposition (e.g., canonical) that we discussed earlier in the section (e.g., sub-section 7.7.5), genealogical justifications can then be asserted to the GTMS, e.g., one per slice.

The disjoint sub-languages special case is also useful for safeties of updating: recall Corollary 6.3. By decomposing before and after an update, this safety by differential decomposition is emergent in the GTMS, in the manner we discussed in section 7.5 and sub-section 7.7.5.

Other implications of the disjoint sub-languages Theorem include locality, selectiveness, and concurrency in backward and forward inference: **recall point 5. in sub-section 7.1.2** and the discussion in section 6.2, including Corollary 6.2.

7.7.10 Conservative Extension

Recall (section 6.4) that any formula E that conservatively extends the previous base is globally monotonic as a base update or as a default update with any priority. This global monotonicity also corresponds to a conjunctive decomposition.

Recognizing when a formula E conservatively extends the previous base may be difficult computationally, in general. Recall, though, that two important special-case conditions suffice to ensure conservative extension: being an explicit definition of a new symbol (Fact 3.7); or being a clause with exactly one literal in a new predicate symbol (Fact 6.9). Each of these conditions can be recognized straightforwardly from surface syntax. In short, procedurally *recognizing* these special cases of conservative extension is tractable: polynomial in the size of the global CLD axiom set.

Procedurally *performing* the global monotonicity and/or conjunctive decomposition is then simple, and can be computed in time linear in the size of the update axiom(s). One or two genealogical justifications can be asserted to the GTMS.

7.7.11 Sympathetically Solitary

The sympathetically solitary special case (section 6.5) has a number of nice properties. It yields a conjunctive decomposition in the form of an "explicit solution" (defined in Observation 6.13), and safeties.

We show below that this explicit solution leads to **new forward inference procedures**, for the clausal case of sympathetically solitary. Forward inference may be selective or exhaustive (with respect to the NM aspect). These special case inference procedures can thus be **added to the library** of circumscriptive inference procedures available in the GCR architecture.

Essentially, the sympathetically solitary case is recognizable from surface syntax. A subtlety is that the sympathetically solitary case is partially defined in terms of the base being *logically equivalent* to a particular form. In general, logical equivalence can be difficult to compute. However, when the global CLD axiom set's base and default formulas are in **clausal** form, we **observe** that the sympathetically solitary case can be **recognized** procedurally in time

$O(n \cdot log\ n)$

where n is the size of the global CLD axiom set.

A sketch of the **algorithm** for recognition in the clausal case is as follows.

1. The first step is to scan the CLD axiom set for the fixtures. These are tested for: they should fix only predicates[14]. A list of fixed predicates is produced.

2. The second step is to analyze the signs of the appearances of all predicates in the literals appearing in the default formulas' clauses. These are tested for: each predicate should have the same sign in every appearance. A list of the predicates Q appearing in the defaults, and of each predicate's sign of appearance σ, is produced.

3. The third step is to analyze the base formulas' clauses. These are tested for: each clause should be in one of two categories: EITHER

   (a) all appearances of predicates in Q have the same sign σ with which those predicates appear in the default formulas ;OR

   (b) there is exactly one appearance of exactly one predicate Qi from Q; AND
   that appearance has sign $\neg\sigma i$ opposite to the sign with which that predicate appears in the default formulas; AND
   any other predicates appearing are fixed.

   A marking of all these clauses is produced: which category they fall into; and, for those in the second category ((b)), which literal is from Q. In addition, a list is produced of: for each predicate Qi in Q: which second-category clauses mention it: these directly yield $Gi[W]$ in Definition 6.17.

The above algorithm requires only one pass over all the CLD axioms / clauses, doing work local to each axiom / clause plus some accesses to auxiliary list data structures: hence its linear cost.

[14] More generally, all of the algorithms in this sub-section can be generalized to permit some of the functions to vary.

Procedurally **performing** the explicit-solution conjunctive decomposition can then be computed in time upper-bounded by

$O(n^2 \cdot log\ n)$

Performing the explicit solution involves substitutions: $G[W]$ is substituted for σQ. Computing $G[W]$ and and performing these substitutions involves an extra cost at most linear in n over the work in the above recognition algorithm.

Thus, for the clausal case, both recognizing the sympathetically solitary condition and performing the explicit-solution conjunctive decomposition is tractable: polynomial in the size n of the global CLD axiom set. There is an additional subtlety to note, however: the resulting explicit-solution conjunctive decomposition may not be clausal: see discussion below.

Safeties:

The sympathetically solitary case also yields several kinds of safeties.

Sympathetic base or default updates (with any priority) are globally monotonic (Corollary 6.22 and Theorem 6.23). This sympathetic condition can be recognized procedurally in time

$O(u \cdot log\ n)$

where u is the size of the update, in similar fashion to the recognition algorithm given above. Procedurally performing the global monotonicity is simple: a genealogical justification can be asserted to the GTMS, as discussed in section 7.5. This takes constant time.

Prioritization updates are globally redundant (Observation 6.24). Procedurally recognizing a prioritization update is simple: it takes constant time. Procedurally performing the global redundancy is also simple: two genealogical justifications can be asserted to the GTMS, as discussed in section 7.5.

Another safety of updating under base updates is partially, not globally, monotonic: Corollary 6.25. This is via differential decomposition. Thus it can be procedurally exploited (recognized and performed) as follows. Recognize and perform the sympathetic-solitary explicit-solution conjunctive decomposition before and after the update. After each decomposition, assert the appropriate genealogical justifications, one per slice, to the GTMS. The partial-monotonicity safety then emerges within the GTMS, as discussed in section 7.5.

New Special-Case Forward Inference Procedures:

The explicit solution yields generating axioms. (Recall the explanation in sub-section 4.3.1 of what generating axioms are.) These generating axioms thus represent (the results of) **forward inference that is exhaustive with respect to the NM aspect** of reasoning (recall sub-section 4.3.1 for this notion of exhaustiveness).

The above explicit-solution performance procedure comprises, therefore, a new special-case NM inference procedure: **forward, exhaustive in the NM aspect, clausal input, sympathetic-solitary**.

There is a practical difficulty, however: clausal conversion. The generating axioms resulting from the above procedure are *not*, in general, clausal. They can, indeed, be reformulated, logically equivalently, to clausal form. However, this **clausal conversion** reformulation may cost exponential time (and space), in the worst case. This general-case intractability of clausal conversion is a well-known difficult in monotonic theorem-proving. Having the generating axioms be in clausal form is quite helpful for performing monotonic inference using them.

The same problem of clausal conversion arises even in predicate completion [Clark, 1978].

Elsewhere (work in preparation), we show that in an important special case, however, the clausal conversion of the generating axioms can, indeed, be performed tractably. The following additional restrictions, taken together, suffice: no fixed predicates (but all functions are fixed), domain closure, and uniqueness of names. These expressive restrictions are common in applications, in our experience. The condition of no fixed predicates is recognizable easily from surface syntax of the CLD axiom set: in time $O(n)$. Domain closure and the uniqueness of names are typically each represented as a special kind of base axiom, and are then recognizable in time $O(n)$ or less.

Selectively and Concurrently:

For both the forward inference procedures discussed above: the algorithms can be straightforwardly modified to perform the explicit solution selectively and concurrently: by slices or subsets of slices. This saves computational effort.

Chapter 8

Discussion

In this chapter, we discuss several, somewhat miscellaneous, topics. Most of these topics touch upon several of the previous chapters; and most involve discussion of related work.

We sketch how our earlier results imply methods for approximation of NM theories and inference, and contrast these with other approaches to approximation. We discuss how our earlier results ease the task of knowledge acquisition. We compare our approach to axiomatizing policy in circumscription to that of Lifschitz [1988a] and show a number of advantages over his variant of circumscription. We discuss related work by Rathmann [1990] about combining NM theories in his variant of circumscription. We briefly review several other NM formalisms: Default Logic, Poole's formalism, Autoepistemic Logic, and logic programs. We observe that many of our major results in this thesis apply to each of these formalisms.

For reasons of space and focus, in this thesis, we do not attempt a comprehensive review of NM formalisms in their relations to circumscription. That could easily be the subject of a dissertation in itself!

In this chapter, we describe various non-circumscriptive NM formalisms, but only quite briefly. For an intermediate degree of detail, more than here but less than primary-source material: [Geffner, 1992] gives a good, short, comparative introduction to non-monotonic reasoning in databases, logic programs, inheritance hierarchies, Doyle-style truth maintenance systems, Default Logic, and Autoepistemic Logic.

8.1 Approximation

In this section, we discuss how our earlier results imply methods for approximation of NM theories and inference, and compare these to other approaches to approximation.

Approximation is a common strategy in computer science to deal with computational difficulty, for example in monotonic inference (e.g., [Kautz and Selman, 1991]).

Currently, the most common approach to approximation in non-monotonic reasoning (e.g., in practical implementations of logic programs with negation and default inheritance[1] in frame-based AI programming systems) is based on the following inference strategy: consistency-checking is performed only incompletely, either in order to save computational resources or because a complete procedure is not available. For example, in negation by failure (inferring $\neg p$ to prove p), the sub-proof attempt (for p) is performed by some procedure that is incomplete; the search for such a proof (of p) is (sometimes) terminated prematurely, without guaranteeing exhaustive coverage. This approach, of approximation by incompleteness in a consistency-checking sub-proof, typically does NOT maintain soundness with respect to the NM theory. Most current NM inference procedures (e.g., those listed in sub-section 4.2.1 and those discussed in section 7.4) involve (monotonic-logical) consistency-checking sub-proofs. But consistency-checking is difficult and sometimes impossible. For propositional logic (or first-order logic with domain closure), consistency is NP-hard. For full (non-monadic) first-order logic, consistency is undecidable. Thus it is understandable that the demands of practicality result in NMR program designers usually settling for approximation by incompleteness of consistency-checking.

Many of our results in this thesis can be viewed as methods for approximation: especially those about decomposition, safety, and design of reasoning procedures (section 4.5, chapters 5 through 7). Our approach to incomplete approximation, by contrast to the above, is to preserve **soundness**. A sound but (perhaps) incomplete approximation can be viewed as a "lower bound".

At a very basic, almost trival level: Any subset of (the conclusions comprising) a NM theory $\mathcal{T}$ can be viewed as an approximation to that theory. For example:

1. the conjunctive combination of any subset of $\mathcal{T}$'s slices, in any conjunctive decomposition of $\mathcal{T}$
2. any tier of $\mathcal{T}$, in any serial decomposition of $\mathcal{T}$
3. any set of conclusions in $\mathcal{T}$'s predecessor theory (predecessor in a sequence of updates) that were guaranteed safe after the immediately-previous update that resulted in $\mathcal{T}$, in any course of updating

are each a sound, but perhaps incomplete, approximation to $\mathcal{T}$. (The approximation is indeed incomplete, for example, if the subset of slices in (1.) above is proper.)

We **observe** that our earlier results imply many further methods of approximation, as well; next, we briefly describe some such. Let $\mathcal{C}$ define the circumscription whose theory is $\mathcal{T}$. Each of the following defines a circumscription / theory that approximates $\mathcal{T}$ soundly but perhaps incompletely:

4. adding more fixtures to $\mathcal{C}$ (by Proposition 3.37)
5. adding more guards to $\mathcal{C}$ (by Proposition 5.9)

[1] Recall footnote on page 15.

6. decreasing the prioritization of the pre-order that is globally minimized (or maximized) in $\mathcal{C}$ (by Theorem 2.58) (By "decreasing" the prioritization, we mean deleting some pairs (paths) from the prioritization partial order (dag).)

7. decreasing the prioritization within any pre-order that is guarded in $\mathcal{C}$ (by Corollary 5.45)

8. eliminating base axioms that mention only fixed symbols (by Theorem 3.58 for the case of base updates)

Each of these methods can can be applied not only to a global circumscription, but also to any subset of the **slices** within a conjunctive decomposition. Note that when applied to slices: (4.), (5.), and (7.) above are each a method of **strengthening protection** (protection in the sense discussed in chapter 5). Replacing some of the slices in a conjunctive decomposition by their sound but (perhaps) incomplete approximations results in a modified conjunction of circumscriptions that is a sound but (perhaps) incomplete approximation to the global circumscription / theory/.

A dual kind of approximation is complete but unsound: this can be viewed as an "upper bound". Methods (4.) through (8.) above can each be reversed straightforwardly to define a circumscription $\mathcal{C}$ that approximates $\mathcal{T}$ completely but perhaps unsoundly:

9. deleting fixtures from $\mathcal{C}$ (by Proposition 3.37)

10. deleting guards from $\mathcal{C}$ (by Proposition 5.9)

11. increasing the prioritization of the pre-order that is globally minimized (or maximized) in $\mathcal{C}$ (by Theorem 2.58) (By "increasing" the prioritization, we mean adding some pairs (paths) from the prioritization partial order (dag).)

12. increasing the prioritization within any pre-order that is guarded in $\mathcal{C}$ (by Corollary 5.45)

13. adding base axioms that mention only fixed symbols (by Theorem 3.58 for the case of base updates)

Again, each of these methods can can be applied not only to a global circumscription, but also to any subset of the **slices** within a conjunctive decomposition. When applied to slices: (9.),(10.) and (12.) above are each a method of **weakening protection**. Note that replacing some of the slices in a conjunctive decomposition by their complete but (perhaps) unsound approximations results in a modified conjunction of circumscriptions that is a complete but (perhaps) unsound approximation to the global circumscription / theory/.

In section 5.11, we discussed the potential expressive complexity of the slices arising in canonical conjunctive (and serial) decomposition along prioritization. Approximating in the above fashion can help to overcome expressive complexity difficulties.

The GCR / GNMR architecture discussed in chapter 7 can be applied straightforwardly to exploit approximations of either kind: sound but (perhaps) incomplete, or complete but (perhaps) unsound.

All that is needed, in either case, is to assert a genealogical justification (defined in sub-section 7.5.4, recall) to the GTMS. Let $\mathcal{TL}$ be a lower-bound (sound but perhaps incomplete) approximation to $\mathcal{T}$; and let $\mathcal{TU}$ be an upper-bound (complete but perhaps unsound approximation) to $\mathcal{T}$. Let tl, t, and tu name these, respectively. Then the genealogical justification

$$t \quad \Rightarrow \quad tl$$

says that tl is a lower-bound for t; and, likewise,

$$tu \quad \Rightarrow \quad t$$

says that tu is an upper-bound for t.

The GCR / GNMR architecture is primarily designed and especially suited, however, for sound but (perhaps) incomplete approximation. In particular, a learning agent's stored set of current conclusions is a sound but (typically) incomplete approximation to the NM theory entailed by its stored current global NM axiom set.

Recently, Cadoli and Schaerf [1992] have developed another style of approximation to non-monotonic theories, for propositional default logic and propositional parallel predicate circumscription. Their approach is to lower-bound (or upper-bound) by weakening (or strengthening) the notion of entailment. Entailment is weakened for some subset of the primitive propositions in the base language. Their approach, like ours, generates incomplete approximations that preserve soundness (or, alternatively, unsound approximations that preserve completeness), in contrast to the usual approach (incomplete consistency checking) that we described at the beginning of this section. Otherwise, however, our approach is quite different from theirs.

Finally, Rathmann [1990] studies approximation in a variant of circumscription that is different from ours. We compare his approach to ours in section 8.4.

8.2 Specification and Knowledge Acquisition

In sub-section 4.3.4, we discussed the difficulty of specification and knowledge acquisition for NM theories. In this section, we briefly discuss how our earlier results bear on this issue.

Firstly, relative to previous versions of circumscription, CLD eases the job of specifying a global set of defaults / minimized predicates and an associated global prioritization partial order. CLD enables one to specify these incrementally; especially convenient is that the global prioritization partial order is generated by transitive closure from the prioritization axioms. In section 7.6, we showed how to mechanize this method of specification in CLD, at little computational cost.

Secondly, the notion of modules and composition of prioritization in our generalization of prioritization (sub-section 2.9.3) implies that one tactic for easing the burden of specification is to specify defaults and priorities in groups, and perhaps hierarchically / recursively: via modules.

Thirdly, our results about conjunctive decomposition can be "inverted" to do **conjunctive composition (terminology)**, by which we mean: to specify NM sub-theories (somewhat) independently, using the concepts of declarations and relevant context from sections 4.5 and 5.10, and then to combine these sub-theories conjunctively. For example, NM sub-theories can be specified and combined coherently using our results about canonical conjunctive decomposition along prioritization (e.g., Corollary 5.21) or about conjunctive decomposition by disjoint sub-language partitions (Theorem 6.1).

Rathmann's thesis [1990] also discusses conjunctive composition; this is much of his focus. We discuss his work in section 8.4.

Fourthly, our results about inference and belief revision, throughout this thesis, are applicable to supporting an **interactive**, automated process of specification and knowledge acquisition. As we discussed in sub-section 4.3.4, such a process requires the exploration of which conclusions are entailed by a "draft" axiom-set specification, and by modifications of that "draft", e.g., to add one or more new axioms to the "draft".

8.3 Lifschitz' Approach to Axiomatizing Policy

In this section, we review related work by Lifschitz on axiomatizing policy in circumscription. By "axiomatizing" policy, we mean specifying policy via axioms in the same formal language as that in which the base (of the circumscription) is specified. We observe a number of advantages of our approach (CLD and PDC) over his.

The only previous approach to axiomatizing policy (Definition 2.8) in circumscription is by Lifschitz [1988a]. His approach, which we will call here "**the JPL framework**"[2] **(terminology)**, generalizes his "pointwise circumscription" [Lifschitz, 1987b]. The JPL framework does not directly represent defaults and priorities. And it is not a special case of general circumscription (Definition 2.7).

Next, we briefly **recapitulate** the JPL framework in a bit of detail.

In the JPL framework, one **defines** a circumscriptive theory as (the set of sentences true in all models of) a **conjunction of circumscriptions**, rather than as a single overall circumscription. Each of these individual circumscriptions minimizes one predicate in the base language: there is one conjunct circumscription for each predicate in the base language. (However, this does not mean that every predicate is effectively minimized; see discussion of "paralysis" below.) There is a single overall base sentence, which is used by every conjunct circumscription. A predicate V is used to describe the policies of these conjunct circumscriptions. V stands for "Varied". V takes as arguments the names of predicates in the base language. Accordingly, the name of each predicate in the base language is introduced as an object (0-ary function) symbol into the base language. **Notation:** for a predicate p, we write its **name** as $'p$. V is a first-order predicate *within* the base language. V has a special reserved role, however, in defining the circumscriptive theory.

$V('p, 'q)$ represents that: in the conjunct circumscription where the predicate p is minimized, the predicate q is varied. (Recall the terminology of varying and fixing from Definition 2.7.)

[2]JPL is mnemonic for "Journal of Philosophical Logic paper".

By contrast, $\neg V('p,'q)$ represents that: in the conjunct circumscription where the predicate p is minimized, the predicate q is not varied, i.e., fixed.

The predicate V itself is required never to vary.

Asserting sentences about V, as part of the base, specifies the policy of each conjunct circumscription, and thus of the overall conjunctive circumscriptive theory.

Which predicates are effectively minimized, and with what prioritization, is represented indirectly.

$V('p,'p)$ indicates that the predicate p is indeed effectively minimized. Perhaps an easier way to understand this is by considering its flip side. $\neg V('p,'p)$ indicates that the predicate p is not effectively minimized. When p is fixed during its own minimization, the result is a kind of paralysis, equivalent to not minimizing p at all.

When the minimized predicates appear positively in the base sentence, e.g., for adaptive-form abnormality theories (Definition 3.5), prioritization is represented, roughly, by asymmetry in varying. When the distinct (adaptive abnormality) predicates p and q are both being effectively minimized,

$$V('p,'q) \ \wedge \ \neg V('q,'p)$$

indicates, roughly, that the minimization of p has higher priority than the minimization of q.

Thus the Quaker-Republican example with strict priority (Example 1.5) can be represented in the JPL framework as the following set of base axioms.

$$\forall x. \neg ab1(x) \ \supset \ (Republican(x) \supset \neg Pacifist(x))$$
$$\forall x. \neg ab2(x) \ \supset \ (Quaker(x) \supset Pacifist(x))$$
$$Quaker(Nixon) \ \wedge \ Republican(Nixon)$$
$$V('ab1,'ab1)$$
$$V('ab2,'ab2)$$
$$V('ab1,'ab2)$$
$$\neg V('ab2,'ab1)$$
$$\neg V('Quaker,'Quaker)$$
$$\neg V('Republican,'Republican)$$
$$V('ab1,'Republican)$$
$$V('ab1,'Quaker)$$
$$V('ab2,'Republican)$$
$$V('ab2,'Quaker)$$

More generally, in the JPL framework, each individual circumscription may be a pointwise circumscription: see discussion of pointwise expressiveness below. (Also, function symbols may be varied as well; V can take their names as arguments. However, this is expressively inessential. This expressive inessentiality follows directly from our observations about functions in section 2.3 plus our results on expressive reducibility of the default case to the predicate case in sub-sections 3.3.2 and 3.5.4.)

Next, we **compare** our approach in this paper with Lifschitz' JPL framework.

Like the JPL framework, CLD enables circumscriptive policy to be specified via axioms within the same formalism as the circumscriptive base. Underlying CLD is prioritized default circumscription (PDC).

Our approach has several **advantages**.

Firstly, CLD has the ability to **represent non-layered prioritization straightforwardly**. By contrast, in the JPL framework, there is no obvious way to represent priority, even in the rough style discussed above, when the prioritization partial order is not layered. This rough-style method uses, in effect, Theorems 5.1 and 5.3 (which are Lifschitz' earlier results, recall). We **observe** that this method does not correspond to prioritization beyond the layered case; this follows from our discussion in section 5.6. We **observe** that, using our canonical conjunctive decomposition result for the one-by-one, predicate case (Corollary 5.18, in ordinary form after guard reduction) plus our results on conservative extension by abnormalities (Corollary 3.46), one can indeed represent PDC with non-layered prioritization in the JPL framework. However, to us, this seems awkward and unnatural as a method for *specification*, as opposed to analysis.

Secondly, in CLD, **the concepts of defaults and priorities (precedence) are represented directly**. By contrast, in the JPL framework, everything about the policy is is expressed in terms of varying versus not varying, i.e, fixing. We find the concept of varying much less intuitive for purposes of knowledge acquisition and specification than the concepts of defaults and priorities. In addition to our own experience with specifying NM theories, there is further evidence that supports our view. Many other NM formalisms besides circumscription also use the concept of a default and/or the concept of prioritization-type precedence. And, perhaps even more importantly, so do most applications to date of expressive NMR.

Thirdly, the JPL framework suffers from an extra source of **nasty unsatisfiability** (recall terminology from beginning of section 3.4). By contrast, CLD does not suffer from this extra kind of nasty unsatisfiability. In general, circumscription may be nastily unsatisfiable in case of non-"well-founded"-ness (Definition 3.16): recall Etherington *et al*'s [1985] example discussed in the beginning of section 3.4, and Observation 3.17. However, the JPL framework can be nastily unsatisfiable for an entirely different reason, as well.

In the JPL framework, we **observe** that: whenever there is a cycle of length 2 or more in the V relation, then there exists a (remainder of the) base sentence such that the circumscriptive theory is unsatisfiable, even though the base sentence is satisfiable.

The following is a simple example of this potential for nasty unsatisfiability in the JPL framework. Let p and q be distinct 0-ary predicates, i.e., propositions. Then the set of base axioms

$V('p,'p)$

$V('q,'q)$

$p \vee q$

$V('p,'q)$

$V('q,'p)$

results in an unsatisfiable circumscriptive theory; the conjunction of circumscriptions is equivalent to:

$$(p \vee q) \ \wedge \ \neg p \ \wedge \ \neg q$$

The problem is that the JPL framework permits a kind of incoherence in the policy. Intuitively, $V('p,'q)$ "tries" to make p's minimization have strictly higher priority than q's, while $V('q,'p)$ "tries" vice versa. Formally, this incoherence is reflected in the fact that this example's conjunction of circumscriptions is not equivalent to any single overall prioritized predicate circumscription, i.e., with a single global policy pre-order, defined over the same base language.

CLD is guaranteed to preserve satisfiability in case of universality (Theorem 3.28), e.g., clausality, or in case of quasi-propositionality (Theorem 3.24). By contrast, the JPL framework is not, as the above example shows. [3]

In effect, the JPL framework places a bit more burden than CLD upon the specifier of a circumscriptive theory. In knowledge acquisition / knowledge engineering, care must be taken to avoid cycles (of length 2 or more) in V. In practice, this might not be too big a deal. For example, we observe that this acyclicity might be enforced by including standard, perhaps quantified, axioms about V in the base sentence, perhaps as part of the (modified) formalism. However, the bigger problem is that varying (i.e., V) is not an intuitive concept for knowledge acquisition / specification in the first place, especially for the commonly-occurring non-layered case of prioritization, as we discussed above.

Fourthly, another advantage of CLD is that it specifies a **single overall circumscription**, with a single overall minimized policy pre-order. The PDC class specified by CLD is a direct generalization of predicate circumscription cf. [McCarthy, 1980] [McCarthy, 1986] and of layered-priority predicate circumscription cf. [Lifschitz, 1985]. Slightly less directly, the JPL framework also is such a generalization. However, unlike the JPL framework, PDC is also a direct special case of general circumscription cf. [Lifschitz, 1984]. And PDC is a direct special case of model-preference NM formalisms cf. [Shoham, 1988] and of "preferential logics" cf. [Kraus *et al.*, 1990], each of which have recently received considerable attention in the NMR literature in the last 5 years. Thus results about these other classes apply more directly, mathematically, to CLD and PDC than to the JPL framework.

However, the JPL framework has two representational features that are not present in CLD. It is able to express monotonic **reasoning about policy**. (V may appear in quantified implications, for example.) And it is able to express **pointwise** policy, i.e., pointwise minimization. Indeed, these expressive capabilities were the main point of developing the JPL framework. In [Grosof, 1992a], we show how to generalize our approach to include both of these expressive capabilities, while retaining all of its above-discussed advantages over Lifschitz' JPL framework. We show there how to expressively generalize further: to allow *non*-monotonic reasoning about policy. CLD corresponds to a special case of this Defeasible Axiomatized Policy (DAP) circumscription. More

[3] More precisely: In the above example, the base sentence is universal. Moreover, the base language is effectively propositional, as well. All but the V part of the base language is propositional. V appears only in ground (base) formulas; it can reformulated as propositional, without doing violence to the example.

precisely, CLD is a syntactic variant of a special case of DAP circumscription.

8.4 Rathmann's Approach to Combining NM Theories

In this section, we discuss related work by Rathmann in his PhD thesis [1990], and compare it to ours in this thesis.

Rathmann addresses issues of specification and inference for large-scale theories in a variant of circumscription. He is especially concerned with how to partition and combine theories so that they may be specified and used in pieces. He develops a framework to do so. Rathmann's work and ours were developed independently.[4]

It is difficult to compare Rathmann's technical results directly to ours, because he employs a significantly different variant of circumscription as his NM logical formalism. This variant treats equality differently from the standard definitions of circumscription by McCarthy [1980] [1986] and Lifschitz [1984] [1985] — and thus differently from the definition of general circumscription used by us in this thesis (Definition 2.7). Its treatment of equality is also different from Lifschitz' pointwise circumscription [1987b] and "JPL framework" [1988a] (discussed in section 8.3, recall), as well as from our definition of prioritized circumscription in [Grosof, 1991]. Rathmann's variant has a semantics that is defined via homomorphisms on models.

Rathmann's motivation and conceptual approach is similar to ours, to a considerable degree. Like us, he wants to break up the problem of inference in an overall NM theory. Like us, he aims to do this by, first, doing inference within each of a collection of NM sub-theories, where inference within each of these NM sub-theories is sound but (perhaps, and typically) incomplete with respect to the overall NM theory; and, second, combining conjunctively these NM sub-theories' conclusions. Like us, he builds on results by Lifschitz, notably Theorem 5.1.

However, there are several important differences in Rathmann's focus and approach from ours. Firstly, much of his focus, and many aspects of his results involve issues revolving around his new formalism, i.e., treatment of equality. Secondly, he is not concerned much with completeness in the process of breaking up: i.e., with whether the conjunction of the pieces is indeed equivalent to the referent global theory. Thirdly, much of his focus, and many aspects of his results, concern *specifying* a NM theory directly as a conjunctive collection of NM sub-theories, i.e., what we call

[4] **A Historical Note:** Rathmann's approach / results and our approach / results were developed independently of each other's. Chronologically, Rathmann developed his approach and results after we developed our conceptual strategy and main results about decomposition, fixing, and generalizing prioritization: in sections 4.3 through 4.6, chapter 5, section 3.5, and chapter 2. This strategy and results of ours were presented publicly in a series of sessions of the Stanford University Non-Monotonic Reasoning Seminar in spring of 1987, and were then circulated in draft form in each of the years since then. However, Rathmann and I did not know about each others' dissertation topics and work until the winter of 1989-1990. This is a bit surprising considering we were both in the same PhD program: Stanford University Computer Science. The probable explanation is that I was not in residence during the period when Rathmann became interested in the topic and did the bulk of his research – I was at IBM T.J. Watson Research Center in NY.

conjunctive composition (recall section 8.2). He considers issues of inconsistency and incoherence that may arise in practice when trying to specify in this manner. Fourthly, he does not consider series circumscriptions or serial combination / (de)composition of NM theories. Overall, Rathmann is concerned primarily with providing semantics for combining NM sub-theories, which he calls "partitioned knowledge bases". He is less directly concerned with developing reasoning procedures than we are, though aiding that is part of the ultimate goal of his work, of course: like us, his aim is to provide theoretical groundwork.

Overall, Rathmann's work and ours are much more complementary than they are competitive.

Next, we compare, in a bit more detail, the part of his work that is most similar to ours reported in this thesis.

Equality issues aside, Rathmann considers the layered-prioritized predicate class of circumscriptions, cf. [Lifschitz, 1985]. His notion of conjunctive combination of "partitioned knowledge bases" is, mainly, properly subsumed by our notion of conjunctive decomposition: he considers soundness of the sub-theories, but does not focus on completeness (as we mentioned above). His focus is more on lower-bounding approximation, in terms of section 8.1.

His main relevant results that are comparable to ours are in his chapter 3: they are about lower-bound approximations. Several of these results are analogous to special cases of, in our terms:

- adding fixtures
- decreasing prioritization
- deleting base axioms that mention only fixed symbols

That is, they are analogous to special cases of approximation methods (4.), (6.), and (8.) in sub-section 8.1.

8.5 Default Logic and Poole

In this section, we briefly review Default Logic and Poole's NM formalism, and discuss how our results apply to them.

Reiter's [1980] Default Logic (DL) is one of the most historically important NM formalisms. A Default Logic theory is specified by two kinds of premise beliefs: a conjunctive collection (set) W of for-sure first-order sentences; and a set D of defaults. W plays a role very similar to that of the base axioms in circumscription, e.g., CLD. Each default in Default Logic is a tentative rule of inference of the form:

$$\frac{\alpha(x)\ :\ \beta(x)}{\gamma(x)}$$

where $\alpha(x)$, $\beta(x)$, and $\gamma(x)$ are open first-order formulas, with a (possibly empty) tuple x of free variables, called the prerequisite (or, sometimes, precondition), the test condition, and the consequent, respectively. These formulas are in the same first-order base language $\mathcal{L}$ as the formulas W. In our usual notation, we can write these alternatively as: $A[Z,x]$, $B[Z,x]$, and $C[Z,x]$, where Z is the tuple of all predicate and function symbols in the base language $\mathcal{L}$. For a tuple t of ground terms, such a default rule allows one to derive $\gamma(t)$ from $\alpha(t)$, provided that $\neg\beta(t)$ is not derivable, i.e., provided that $\beta(t)$ is consistent in a particular sense. When $\beta(x)$ and $\gamma(x)$ are the same, the default rule is called *normal*. When x is empty, the default rule is called *closed*; the default is then propositional. Generally, an open default rule is interpreted as a schema; i.e., it is defined as equivalent to the collection of all its instances, each defined by a ground t.

For example, let W consist of the single sentence $Republican(Nixon)$, and let D consist of a single (normal) default

$$\frac{Republican(x)\ :\ \neg Pacifist(x)}{\neg Pacifist(x)}$$

Then $\neg Pacifist(Nixon)$ is a NM conclusion.

However, if W also includes $Pacifist(Nixon)$, then the default is violated (for the $Nixon$ instance). That is, default conclusion is blocked: the test condition fails, i.e., is inconsistent. The former NM conclusion no longer holds.

DL defaults may also conflict. For example, let W also include $Quaker(Nixon)$, instead of $Pacifist(Nixon)$, and let D also include the (normal) default

$$\frac{Quaker(x)\ :\ Pacifist(x)}{Pacifist(x)}$$

Then the two default rules conflict: the prerequisites of both defaults (more precisely, of their $Nixon$ instances) are satisfied, but application of one default blocks the other and vice versa.

Reiter handles conflict by introducing the notion of *extensions* of a default theory. In general, conflict between defaults leads to multiplicity of extensions. Each extension is a candidate set of NM conclusions, i.e., a candidate NM theory, in our terms. Note that what Reiter calls a "default theory" is closer to what we call a NM axiom set: it is a specification of premise beliefs, rather than a collection of NM conclusions.

An extension E of a normal "default theory" $\langle W, D\rangle$ is a minimal deductively closed set of formulas E such that: 1) $W \subseteq E$; and 2) every default $\frac{\alpha\ :\ \beta}{\gamma}$ in D whose prerequisite α is in E is either blocked, i.e., $E \models \neg\beta$, or has its consequent γ in E. More generally and formally, Reiter defines an extension of a (not necessarily normal) "default theory" as the fixed point of an iteration in which default rules are applied sequentially. For normal "default theories", this iteration has the flavor of a serial combination cf. serial decomposition, similar to Theorem 5.69.

In the last example above, there are two different extensions: one which corresponds to the Republican default winning on the Nixon instance, and another which corresponds to the Quaker default winning on Nixon. The first extension includes the NM conclusion $\neg Pacifist(Nixon)$, while the second extension includes the NM conclusion $Pacifist(Nixon)$.

Recalling our discussion of skeptical versus credulous on page 23., Default Logic is a **credulous (terminology)** NM formalism: in general, there may be several, mutually exclusive, alternative

NM theories (i.e., the extensions) that are sanctioned. By contrast, circumscription is skeptical: only a single body of NM conclusions is sanctioned.

Note, furthermore, that Default Logic has no means of specifying prioritization-type precedence.

A **skeptical version** of Default Logic is defined straightforwardly. Viewing an extension / NM theory as a set of conclusions: the skeptical NM theory is defined as the intersection of all the multiple (credulous) extensions. Viewing an extension / NM theory as a conjunction of conclusion sentences, i.e., as an axiomatization: the skeptical NM theory is defined as the disjunction of all the multiple (credulous) extensions. Viewing an extension / NM theory in terms of first-order-logical models: the set of models of the skeptical NM theory is defined as the union of the sets of models of all the multiple (credulous) extensions.

Semi-Monotonicity of Default Updates in Default Logic: One of the only previous results on belief revision for default updates in an expressively rich NM formalism is by Reiter [1980] Theorem 6.3 for closed, normal Default Logic. His result is that: after updating with a new default, the new set of extensions contains an extension that includes one of the previous extensions. I.e., there exists a new extension which is a (non-strict) strengthening of one of the old extensions. Reiter calls this property *semi-monotonicity*. We remark that this semi-monotonicity is a fairly weak result for learning agents, however, since the formalism here is credulous. For the skeptical NM theory, it does not guarantee anything about the (even partial) monotonicity of the update.

Quite a lot is known about the relationship of Default Logic to circumscription [Grosof, 1984] [Imielinski, 1987] [Etherington, 1987] [Etherington, 1988] [Lifschitz, 1990b] [Qian and Irani, 1991]. Most relevant for our purposes is that there is a rough equivalence between the maximization of a default formula (e.g., the the minimization of a predicate) in circumscription, and a normal default without prerequisite in Default Logic. (By without prerequisite, we mean that the prerequisite is empty, i.e., $True$.) The following observation makes this more precise.

Observation 8.1 (Overlap between Default Logic and Circumscription)

Suppose the for-sure beliefs B entail: 1) domain closure (Definition 2.23); and 2) a complete theory of equality (Definition 2.22), e.g., uniqueness of names (Definition 2.21).
Then, for any set of default formulas G, indexed by N:
the parallel default circumscription without explicit fixtures $PDC(B; G^N; \emptyset; Z)$
is equivalent to the skeptical version of the DL default theory $\langle B, D\rangle$,
where each DL default $Di \in D$, for $i \in N$, is defined as:

$$\frac{:\ Gi}{Gi}$$

By equivalent here, we mean that the NM theories are the same.

Note that the condition 2) implies that the base language is effectively propositional.

Note also that it does not matter whether or not functions are fixed in the circumscription, by Theorem 2.24.

Proof : **Sketch:** This follows straightforwardly from Etherington's [1988] Theorem 8.3, which is for the predicate case. To generalize the predicate case to the default-formula case, apply conservative extension for definitional-style abnormalities (our Theorem 3.8 or Lifschitz' [1984] Proposition 7) to the circumscription, and apply its analogue in Default Logic to each extension. □

In short, an important class (i.e., effectively-propositional, normal, without prerequisite, skeptical version) of Default Logic is equivalent to parallel default / predicate circumscription. We **observe** that, therefore, **most of our main results for parallel PDC apply to this case of Default Logic: notably those about decomposition, safeties of updating, the GNMR architecture, and abnormalities**, in sections 3.3, 4.5, 5.1 through 5.11, and all of chapters 6 and 7. Especially interesting in this connection, for example, are our results on canonical decomposition (including the concept of a justifying set of defaults; recall discussion in section 5.8), decomposition and updating in sympathetic-solitary theories, the GTMS, and the GNMR architecture. See sub-section 9.1.2 for a bit more detailed discussion of which results carry over.

Poole's Logical Framework for Default Reasoning:
Poole [1988] defines a NM formalism that is equivalent to a special case of Default Logic: where defaults are restricted to be normal without prerequisite. We **observe that our results thus carry over to Poole's formalism** as well, in a similar fashion as we discussed above for Default Logic. Indeed, our results are even more particularly applicable to Poole's formalism than to general Default Logic.

8.6 Autoepistemic Logic

In this section, we briefly review Autoepistemic Logic, and discuss how our results apply to it.

Moore's [1985] Autoepistemic Logic (AEL) is another of the most historically important NM formalisms. Moore developed it as a reconstruction of McDermott's and Doyle's [1980] NM modal logic.

Autoepistemic Logic augments propositional classical logic with a modal operator L, where a sentence of the form $L\alpha$ is to be read as "α is believed". "Belief" here has a very particular sense.

Like Default Logic, Autoepistemic Logic is credulous, and has no means of specifying prioritization-type precedence. In AEL, the correspondent of a DL extension is a *stable expansion*. The stable expansions of a base theory B, where B is a set of sentences in this modal language, are defined as the sets of modal sentences $S(B)$ which satisfy the fixed-point equation:

$$S(B) \quad = \quad Th(B \cup \{Lp \mid p \in S(B)\} \cup \{\neg Lp \mid p \notin S(B)\})$$

where, for any T, $Th(T)$ stands for the the set of all sentences tautologically entailed by T. Stable expansions are thus closed under both positive and negative introspection.

In a similar fashion to Default Logic, a **skeptical version** of Autoepistemic Logic is defined straightforwardly. Viewing a stable expansion / NM theory as a set of conclusions: the skeptical NM theory is defined as the intersection of all the multiple (credulous) stable expansions.

As Konolige [1988a] and others since have shown, there is a close relationship between Autoepistemic Logic and Default Logic; and, thus, by transitivity, between AEL and circumscription.

Most relevant for our purposes is the following. Consider a default axiom, with formula part Di, in propositional CLD without priorities or fixtures. This corresponds roughly to asserting the modal sentence

$$L\neg Di \supset \neg Di$$

as part of the base theory B in the skeptical version of AEL.

Lifschitz [1989] gives another, interesting relationship between AEL and circumscription, via a new, "introspective" variant of circumscription.

Thus we **observe that most of our main results are applicable to the skeptical version of Autoepistemic Logic**, just as they are to the skeptical version of Default Logic: via what is known about equivalences between AEL and circumscription, including transitively via Default Logic. There is a subtlety, however. Recall that our definitions of conjunctive and serial decomposition were developed in terms of classical logic as the "base logic". To apply them to AEL, we need to adapt them to the *modal* language / monotonic logic associated with AEL, instead of *classical* language / monotonic logic. This is straightforward: see the footnotes in Definitions 4.2 and 4.11 for where this modification needs to be made.

8.7 Logic Programs

In this section, we briefly review logic programs, and discuss how our results apply to it.

A logic program is a collection of implicitly universally quantified rules βi, for $i = 1, \ldots, m$, $m \geq 1$, of the form

$$Ai \;\leftarrow\; Li1, Li2, \ldots, Lin$$

Here Ai is an atom, called the "head" of the rule. Each Lij, for $ij = 1, \ldots, n_i$, where $n_i \geq 0$, is a positive or negative literal in the rule's "body". (Typically, these literals are in classical first-order logical syntax.) Logic program rules take both a procedural and a declarative reading. The usual understanding of negation in this context is: negation by failure, i.e., negation by failure to prove [Clark, 1978]. Giving this negation-by-failure a declarative meaning must be non-monotonic; making this meaning precise is, in general, a thorny challenge. For the by-now-standard **stratified** class of logic programs [Apt *et al.*, 1987] [Van Gelder, 1988], however, there currently exists a standard, near-consensus declarative interpretation. **Lifschitz** [1987a] [1988b] (see also [Przymusinski, 1988]) **shows** that, under this declarative interpretation, a stratified logic program is **semantically equivalent to a special case of prioritized predicate circumscription**:

$$PPC(B; Z; S; Z)$$

Here,

$$B \;\stackrel{\mathrm{def}}{\equiv}\; \bigwedge_{i=1}^{m} Bi$$

where Bi is the result of replacing $\leftarrow$ in βi by material implication:

$$Bi \;\stackrel{\mathrm{def}}{\equiv}\; [(Li1 \wedge Li2 \wedge \ldots \wedge Lin_i) \;\supset\; Ai]$$

In this circumscription, **all of the predicates (Z) in the base language are minimized** according to a **layered prioritization** partial order S that corresponds to a *stratification* of the predicates. This stratification is defined (non-deterministically) in terms of which predicates appear in negative literals. The essence of stratification is the constraint that a predicate appearing in the head of a rule must be in a lower-priority layer (stratum) than any predicate appearing in a negated literal in the body of that rule.

We **observe**, therefore, that **many of our main results apply to stratified logic programs** with this declarative semantics. Particularly interesting in this regard are: the safety of higher-priority conclusions (section 5.12), conservative updates (6.4), and the use of decomposition, the GTMS, and the GNMR architecture, especially to support forward inference and concurrency.

8.8 Brewka's Preferred Sub-Theories

Brewka [1989a] [1989b] extends Poole's [1988] formalism (discussed in the last section) by adding prioritization-type precedence, and, more generally, extends Default Logic likewise. **Recall** our discussion in sub-section 2.11.3: his notion of prioritization is somewhat different than ours, however. In this section, we give an example showing the difference.

Brewka introduces precedence into Poole's formalism by definitionally constraining the permitted sequences of application of defaults: only those sequences are allowed that are descending (Definition 5.31) with respect to the precedence partial order on the defaults.

Example 8.2 (Counterexample to Equivalence with Brewka)
This example is **due to Andrew Baker** (private communication, 1989). Let the defaults D be a tuple of distinct primitive propositions: $\langle p1, p2, p3, p4\rangle$. Let the prioritization partial order relation R consist exactly of the two pairs $(1,2)$ and $(3,4)$. I.e., $p1$ is maximized at higher priority than $p2$, and $p3$ is maximized at higher priority than $p4$. Note that this prioritization partial order is non-layered, and columnar as in Figure 2.4. Let the for-sure beliefs B be:

$\neg(p1 \wedge p3)$

$\neg(p1 \wedge p4)$

$\neg(p2 \wedge p3)$

(Let there be no explicit fixtures.)

Then the skeptical NM theory in Brewka's formalism is equivalent to:

$$(p1 \wedge p2 \wedge \neg p3 \wedge \neg p4) \;\vee\; (\neg p1 \wedge \neg p2 \wedge p3 \wedge p4)$$

This is strictly stronger than the prioritized default circumscription

$$PDC(B; D; R; p)$$

defined by our notion of prioritization (Definition 2.33). This PDC is equivalent to:

$$(p1 \wedge p2 \wedge \neg p3 \wedge \neg p4) \;\vee\; (\neg p1 \wedge \neg p2 \wedge p3 \wedge p4) \;\vee\; (\neg p1 \wedge p2 \wedge \neg p3 \wedge p4)$$

In Brewka's formalism, every permitted sequence begins by maximizing either $p1$ or $p3$. Thus every extension contains either $p1$ or $p3$. However, in prioritized circumscription, the model $M3$ corresponding to

$\neg p1 \wedge p2 \wedge \neg p3 \wedge p4$

is not strictly less preferred than the model $M1$ corresponding to

$p1 \wedge p2 \wedge \neg p3 \wedge \neg p4$

nor than the model $M2$ corresponding to

$\neg p1 \wedge \neg p2 \wedge p3 \wedge p4$

Though $M3$ is "worse" on the $p1$ default than $M1$, it is "better" on the $p4$ default, and there is no strict priority between the (maximizations of) $p1$ and $p4$. Likewise, though $M3$ is "worse" on the $p3$ default than $M2$, it is "better" on the $p2$ default, and there is no strict priority between the (maximizations of) $p2$ and $p3$.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

Guide to Reader: Recall the thesis abstract and the discussion of problems and methodology in sections 1.1, 1.2, and, especially, 1.3; this section is meant to complement them. Note that our discussion of future work (section 9.2) provides additional context for the conclusions.

For more detail than this section, see also the overview material in the previous chapters, and the thesis outline in section 1.4.

9.1.1 Summary of Novel Contributions

In this sub-section, we summarize the novel contributions of this thesis. We aim to provide an intermediate level of detail: more detail than the thesis abstract and section 1.3 (nature of results), but less detail, plus more unifying coherence, than the overview material in the individual chapters.

For the sake of brevity, our format below is fairly terse, similar to an extended abstract. We use the LaTeX sans serif font style to flag terminology that is particular to this thesis, relative to the NMR literature.

This thesis has the taken the form of a theoretical, groundwork study. We addressed the problems of non-monotonic reasoning (NMR): specification, inference, and, especially, updating. We were motivated especially by applications in the realm of empiricism and action. We developed the concept of a learning agent as an abstraction of this kind of NMR. These applications involve large scale, plus expressive generality. We analyzed the kind of expressive generality demanded: fairly arbitrary partial orders of precedence among defaults, as well as fairly arbitrary first-order forms of default and for-sure beliefs. We focussed on two goals, therefore. The first was to attack the problem of scale for expressively rich NMR. The second was to develop expressive generality with respect to precedence.

We laid seige to the problem of scale at the logical level, which is relatively independent of application domain. However, an important limitation in principle of this work is, therefore, that we could not say anything directly about the effects of resource constraints on reasoning, i.e., about "resource-bounded reasoning".

We concentrated on tackling the difficulty of NM reasoning that is due purely to non-monotonicity, i.e., the difficulty that goes beyond that of monotonic reasoning. We analyzed the problem of scale in terms of logical globality, which is especially manifested in the belief revision aspect of updating. We developed a divide-and-conquer strategy of attack on logical globality and belief revision. Key: We developed concepts of hierarchical (recursive) conjunctive and serial decomposition, applicable beyond circumscription, and showed their potentially-in-principle pay-offs: (guarantees of) localities in belief revision and inference, concurrency in inference, and selectiveness in forward inference. Part of our strategy is to use decomposition results to prove other kinds of localities: especially, to prove safeties of updating by taking the differential of decomposition before and after an update. We developed the concept of guarding to represent the structure of potential conflicting interaction in circumscription. We observed an analogy between decomposing NM theories and decomposing programs in programming languages with side effects. In this analogy, guards play the role of declarations of non-local variables written / read.

We also developed concepts of safety behavior under updating, applicable beyond circumscription: partial and global, forever and current, redundant and monotonic. We analyzed some of the difficulties of belief revision in circumscription, in terms of trampled properties.

We treated updating with new defaults and new priorities, not just with new for-sure axioms. We observed, and illustrated with examples, that this especially important to consider because in many applications, especially for learning agents, relatively little of the information that one acquires dynamically, e.g., through perception and communication, is for-sure: most is defeasible with some degree of confidence.

We chose the circumscription formalism as our vehicle for analysis for several reasons. It is historically central, relatively well-studied previously, mathematically close to classical logic, and meets our main expressive concerns: it can express precedence, as well as defaults and for-sure beliefs of arbitrary first-order form, and is skeptical. However, circumscription lacks some expressive features offered by various other NM formalisms, e.g., in treatment of antecedent versus consequent in default rules, or of introspection.

We improved the expressiveness of the circumscription formalism: we generalized the definition of prioritization beyond layered: to arbitrary finite, and well-founded infinite, partial orders of precedence. We showed that this generalization is indeed important for default reasoning, e.g., to represent adequately the kind of precedence found in inheritance networks or, more generally, from specificity dominance. We observed that two additional bases for precedence, little remarked upon previously in the NMR literature, are common and important: source reliability; and authority in the legal and organizational senses. We illustrated these with examples and showed that they too demand non-layered-ness.

We developed the idea of composing prioritization recursively / hierarchically into modules, and showed that this is well-defined. We showed, by example, that this idea of prioritization as structure among default axioms is often a good match to the kind of premise information naturally available: e.g., columns in inheritance; topics; and sources.

Our idea of prioritization applies beyond circumscription. We showed that it subsumes several previous ideas in the NMR literature, and contrasted it with others. Indeed, our idea of prioritization applies beyond even NMR: it is a notion of aggregation of preference criteria. We extended previous desiderata for such aggregation, and justified our definition in those terms.

We improved the expressive convenience / natural-ness of the circumscription formalism by developing the Circumscriptive Language of Defaults (CLD) formalism, in which defaults and priorities are specified directly, and incrementally, as axioms. This directness and incrementality also make CLD more suitable than previous variants of circumscription as a vehicle to study updating and learning agents. Many of our decomposition and safety results manifested similarity between the behavior of default axioms and that of base axioms. This similarity is obscured when everything is described in terms of minimizing predicates.

CLD is syntactic sugar: it is a meta-language for specifying prioritized default circumscriptions. We showed that prioritized default circumscription (PDC) expressively reduces to the abnormality-style case of prioritized predicate circumscription (PPC). We sketched how to implement this reduction reformulation automatically as a DP-converter, and observed that it imposes only reasonable, polynomial cost.

PPC is a special case of PDC. Predicate circumscription has been more studied previously than default circumscription and several previous inference procedures are available for it. We developed query-answering inference procedures for (effectively-propositional) parallel and layered PDC, by piggybacking our DP-converter on top of some of these previous inference procedures for PPC.

One expressive convenience of CLD is that the transitive closure of the specified prioritization axioms is performed implicitly; we sketched how to implement this as a CLD interpreter and observed that it imposes only reasonable, polynomial computational cost.

We formalized the previous usual convention of usage of abnormalities and showed that the folkloric expectations underlying that usage are borne out even when there is non-layered prioritization.

We analyzed basic well-behavior for our new expressive classes (non-layered and default). In particular, we showed, for CLD / PDC / PPC, that the circumscription is guaranteed to preserve satisfiability when either of two conditions hold: universality; and quasi-propositionality. We observed, reassuringly, that these conditions are commonly met in practical AI knowledge representation. Special cases of quasi-propositional include: 1) propositional; and 2) domain closure with a complete theory of equality.

We showed a collection of results about localities of NMR in prioritized default circumscription, from quite general to fairly special cases. Key: We showed our decompositional strategy to be, indeed, fruitful: in particular, we often used decomposition results to prove safeties of updating. We also used decomposition results to prove new inference algorithms, and opportunities for selectiveness and concurrency in inference.

One basis for general-case decomposition is prioritization. We showed that *any* PDC is canonically conjunctively decomposable along its prioritization, into one slice per default. This includes the case where prioritization is empty, i.e., parallel. We showed also that any PDC is, likewise, canonically *serially* decomposable into one phase per default, in sequence that is descending with respect to the prioritization. These canonical decompositions use a canonical protection tactic, which is the same for both conjunctive and serial decomposition. In addition, we showed that any quasi-propositional PDC is canonically serially decomposable one-by-one in *every* sequence, even in reverse prioritization sequence.

We showed, moreover, that decomposition can be performed hierarchically, at a variety of grain sizes from macro to micro: by decomposing the prioritization partial order itself, thus taking advantage of our idea of modules. We showed that any quasi-propositional PDC is canonically decomposable into one slice or phase per default *instance*. This finest-grain decomposition eliminates the possibility of conflict between defaults within the resulting sub-theories.

We showed also that canonical decomposition goes beyond the default case: it applies to any prioritized circumscription.

Canonical protection introduces fixtures. We analyzed fixing's direct and indirect sources, and showed a number of its effects, including behavior under updating. The most central intuition behind fixing is the immunity of fixed formulas to the non-monotonic effect of the circumscription operation. We showed well-behavior conditions under which this intuitive characterization actually holds for non-layered PDC, including: universality and quasi-propositionality.

We found the non-layered case of priority to be much more complicated to decompose than the layered case. We showed the necessity, in general, of the canonical protection conditions. We analyzed the expressive complexity of the canonical slices and phases. Worst-case, this may involve an increase relative to the original global axiom set: squared blow-up of the CLD representation, and loss of clausality. In many cases, however, the expressive complexity of the slices and phases is clearly simpler; we investigated some of these special cases (see below).

Researchers often speak intuitively of deriving a NM conclusion from a single default or, more generally, from a proper subset of the global defaults. We formalized this for PDC by developing concepts of canonical derivability. In doing so, we used the canonical protection tactic, and thereby skirted the unsoundness of a simplistic approach.

We used canonical decomposition to show the safety of higher-priority conclusions under default updates or, more generally, module updates; our method was by differential decomposition. We showed, by example, that defining precisely which are these higher-priority conclusions involves subtleties when prioritization is non-layered: our solution is to use canonical derivability.

Our canonical decompositions imply opportunities for selectiveness and concurrency in inference. Being general-case, however, our canonical decompositions are most useful when combined with special-case results.

We showed results about strong localities available in special cases. We used the general-case decompositions to help prove these special-case results. Strong locality is manifested in clean decomposition, in which defaults are guaranteed not to conflict at all between sub-theories.

One special case is total prioritization over modules: here we showed strong serial decomposition. We used this case to prove that base axioms are equivalent to strictly-highest-priority defaults. Another, even more special case is totally-prioritized propositional defaults: here we showed a new exhaustive forward inference procedure as well, which includes the ability to be selective without sacrificing soundness.

Other special cases are defined primarily in terms other than prioritization: orthogonality and sympathy. Priority dominance corresponds to the concept of winning in battle. The concept of logical orthogonality corresponds to the intuition of passing by on different geometric planes, or of "ships" passing in the night. More formally, orthogonality is a kind of logical independence.

One case of logical orthogonality is disjointness of mentioned predicate (and function) symbols. We discovered non-trivial subtleties, however: prioritization, especially non-layered prioritization, and fixture each complicate matters. We defined a generalized disjoint sub-languages condition and showed that it ensures clean conjunctive decomposition. We used this decomposition to prove locality of backward and forward inference, opportunities for selectiveness and concurrency in inference, and powerful safeties of updating.

Most large axiom sets of interest for applications, however, do not display much perfect partitionability of symbols. The real power of our disjoint sub-languages results comes when they are combined with a kind of definitional reformulation. Then a much broader class of disjointly describable axiom sets can be shown to have clean slicings. Sub-section 9.2.1 discusses how we have exploited the disjoint sub-languages results in our follow-on work.

Another case of logical orthogonality is default or base updates whose formula part conservatively extends the previous base, e.g., when the update is a rule mentioning a new predicate symbol. Here we showed the update to be globally monotonic; this corresponds in turn to a simple case of decomposition.

The concept of sympathy corresponds to the intuition of pulling in the same direction, or working in harmony. Technically, we used the ideas of syntactic positivity and negativity to capture the notion of directionality.

One case of sympathy is sympathetic-solitary, in which default reasoning is restricted to be shallow in a particular sense. We discovered this case by deliberately setting out to look for a case in which non-conflict can be guaranteed at finest-, i.e., instance-, grain. For the sympathetically solitary class, we showed a first-order explicit solution to the whole circumscription. We used this solution to show powerful safeties of updating, and clean decompositions at a range of grain sizes. And we used it to develop an exhaustive forward inference procedure for sympathetic-solitary: with opportunities for selectiveness and concurrency.

Another case of sympathy is strong sympathy, in which new base or default formulas are formed purely and positively from previous default and fixed formulas, and prioritization is parallel. We showed such strongly sympathetic updates to be globally and forever monotonic; this corresponds in turn to a simple case of decomposition.

In addition, we showed that updating by adding prioritization, e.g., in CLD, is globally monotonic.

We showed, with a long running example, that our general-case and special-case decomposition and safety results are synergistic and can be applied hierarchically. Many different decompositions can be applied to the same NM theory. E.g., one decomposition result can be used to slice within the phases resulting from another decomposition result. And different kinds of safety guarantees can be applied in the course of updating a single theory.

We observed that our decomposition and safety results also imply methods for approximation and for interactive knowledge acquisition.

We discussed related work, and observed advantages of our approach (CLD) to axiomatizing policy in circumscription over the previous work by Lifschitz.

We discussed how to exploit our results to design procedures for NM specification, inference, and belief revision: primarily, in PPC and PDC; and, more generally but to a lesser extent, in other NM formalisms as well. We sketched an architectural method for sound (but perhaps, and typically, incomplete) NMR, especially suitable for learning agents. This architecture is able to exploit *any* results, e.g., ours, on decompositions and safeties, as well as any inference procedures. We call this architecture "Genealogical" because conclusions descend to the current global theory from slices, tiers, and previous theories.

We developed in detail the Genealogical Truth Maintenance System (GTMS): the book-keeper at the heart of this GNMR architecture. We showed the GTMS task to be computationally tractable: polynomial in the justifications, unlike the ordinary ATMS task, which is NP-hard.

We sketched how to implement most of our major results on decompositions, safeties, reformulations, and inference procedures. We observed that the computational cost involved, both to recognize the relevant conditions / special cases and to perform the decompositions etc., is polynomial in the size of the global CLD axiom set. This tractability was an outgrowth of part of our strategy: to seek results for cases formulated in terms of CLD surface syntax.

Others of our results, however, primarily give analytic insight, at least for now.

Key: Thus, we observed that most of our major results, including the GTMS and the GNMR architecture, are practically implementable.

The GNMR architecture provides a method to "pick off" well-behaved cases within an expressively general class of NM reasoning. The implementable special cases in our results are also useful as a target for specification and expressive restriction.

We were not able, however, to evaluate the practical usefulness of our approach and results in more detail, e.g., with implemented experiments. That awaits future work.

Finally, we observed that much of our major results and concepts, including about decomposition, safeties, the GTMS, and the GNMR architecture, do indeed apply beyond circumscription to each of a variety of other expressively rich NM formalisms, including: the skeptical versions of Default Logic, Poole's formalism, and Autoepistemic Logic. In addition, we observed that they apply to default inheritance (with examples) and to stratified logic programs too.

9.1.2 Discussion: Some Overall Highlights

In this sub-section, we discuss some of what we view as the most important overall contributions of this thesis, and lessons to take away from it.

Most Conceptually Innovative Contributions

In our view, the single most innovative aspect of our work are the concepts of decomposition and bloc safety, and the strategy of attack on (the NM aspect of) inference, especially forward inference, and belief revision / updating that we built around it.

Another of the most innovative aspects of our work is the emphasis on updating with new priorities and new defaults. There had been very little previous work on monotonicities of updating with new defaults, and none with new priorities, in any of the three most-studied and historically central NM formalisms: circumscription, Default Logic, or Autoepistemic Logic, nor in many other expressively rich NM formalisms / systems, e.g., preferential logics cf. Kraus et al.

The Value of Prioritization

Many of our results revolved around prioritization, including how to exploit it for locality. In most previous work on NMR, precedence has not been emphasized. A lesson to take away from this thesis is that prioritization information is valuable and analytically important, and deserves more attention.

Interestingly, precedence plays a key role in the most expressive kinds of NMR that are currently in most widespread (relatively) practical use: default inheritance and logic programs. In most inference algorithms for default inheritance, e.g., in Touretzky's [1986], inferential processing proceeds bottom-up: from more specific, i.e., higher-precedence, to less specific, i.e., lower-precedence. Likewise, most inference algorithms for the stratified class of logic programs (recall section 8.7) with negation as failure apply their negation-by-failure rule in the direction that corresponds to going from higher- to lower- precedence.

Most Innovative Technical Contributions

What we view as the most innovative technical contributions are:

- canonical decomposition for non-layered priority, and discovery of its subtleties (sections 5.1 through 5.6)
- safety of higher-priority conclusions for non-layered priority, and discovery of its subtleties (section 5.12)
- explicit solution for sympathetic-solitary (section 6.5)
- the GTMS and the GNMR architecture (sections 7.2 and 7.5 through 7.7)

Note that the GNMR architecture can utilize *any* future-developed procedures for NM inference and safety.

Significance In the Absence of Precedence; DL and AEL

Many other NM formalisms are not equipped with the ability to express precedence, e.g., Default Logic and Autoepistemic Logic. For application of our results to (the skeptical versions) of these formalisms, the parallel, i.e., empty-prioritization, case of our results are important. Each of the following is applicable to the skeptical version of Default Logic, cf. Observation 8.1 (section 8.5), for example:

- conservative extension for abnormalities (and fixtures) (Corollary 3.46)
- the new query-answering inference procedure for parallel default circumscription (sub-section 7.6.2)
- canonical conjunctive or serial decomposition (sections 5.4 and 5.5) (Note that the expressive complexity of the canonical slices / phases is not much of a difficulty in the parallel case; recall section 5.6.)
- sympathetic-solitary: explicit solution and safeties (section 6.5); and new forward inference procedures, both exhaustive and selective (sub-section 7.7.11)
- the GTMS and GNMR architecture (sections 7.2 and 7.5 through 7.7)
- conservatively extending updates (section 6.4)
- disjoint sub-languages (section 6.2)
- strong sympathy (section 6.6)

Implications

Some of the most notable, more immediately practical implications of our thesis include:

- guidance as to expressive restrictions that ensure relatively easier NMR
- new NM inference procedures: forward, and exhaustive or selective, for total-propositional (sub-section 5.13.2) and sympathetic-solitary (section 6.5); and backward for layered default circumscription (sub-section 7.6.2)

9.2 Future Directions

In this section, we discuss directions for future work that appear most important, interesting, and likely to be fruitful.

Overall, this thesis provides a theoretical groundwork for designing programs to perform large scale, expressively rich default NMR, especially learning agents concerned with belief revision / maintenance and forward inference. It is a first step and methodology for more powerful results.

We believe that the basic approach, and much of our results, will apply to many other non-circumscriptive NM formalisms, beyond just those discussed in sections 8.5 through 8.7: we list some below.

9.2.1 Expressive Restrictions that Yield Locality

One interesting direction is further theoretical investigation of well-behaved expressive special cases: i.e., that have strong localities. The challenge is to find expressively-restricted cases that are useful for applications. A major achievement would be to find classes more expressively general than current default inheritance or logic programs, but with relatively cheap computational cost for inference and belief revision. For example, a major achievement would be to embed default inheritance within a monotonic logic programming system, at relatively low cost.

Follow-on Work: Individual-wise Disjoint Describability:
Building on this thesis, we have done, elsewhere, some work in this direction. In [Grosof, 1992c], we identify a broad class of prioritized default theories, generalizing default inheritance, with strong localities. For this **asocially monadic** class, decomposition permits reasoning to be localized to individuals (ground terms). Our proof method is by disjoint describability (section 6.3), involving definitional reformulation of a particular kind, as well as clean conjunctive decomposition in the manner of our disjoint sub-languages case (section 6.2). (Recall our discussion of such individual-wise disjoint describability in section 6.3.) The asocially monadic class generalizes default inheritance cf. Touretzky [1986] by permitting negation and limited disjunction in the base and default formulas. We observe there that the asocially monadic class can be recognized, and the individual-wise reformulation can be performed, in time that is polynomial in the size of the global CLD axiom set.

Other Tactical Avenues:
As a means to find useful expressively-restricted cases with strong locality, several other tactical avenues hold promise for strengthening our results. One is to find more cases of logical orthogonality and sympathy. Intriguing in this regard are Rabinov's [1989] results about when layered PPC collapses to first-order. Rabinov's results generalize those previously of Lifschitz about "separable" and "pointwise" circumscriptions. The solitary case (Definition 6.11) is a special case of separable and thus of Rabinov's class. In section 6.5, we generalized the solitary class to sympathetically solitary, which is more useful for applications but retains attractive computational ease. Perhaps

Rabinov's class can be developed further in a like manner.

Another avenue is to generalize serial decomposition to consider tiers that result from incomplete, rather than only complete / exhaustive, NM forward inference on a phase: i.e., to draw a sound NM conclusion, then treat that as for-sure and proceed with further NM inference.

9.2.2 Implementation; Fleshing out the GNMR Arch.

Another interesting direction is to implement, and experiment with, the GNMR architecture. Implementation will involve fleshing out the GNMR architecture, especially its control aspects, as well as detailed algorithm designs for our new procedures of various kinds, including for: inference, safety, the GTMS, the CLD interpreter, the DP-converter, recognition of various cases, etc. (all discussed in sections 7.5 through 7.7, recall). Interactive knowledge acquisition (section 8.2) is an issue in this connection.

Experimentation is needed to evaluate the practical usefulness of our approach.

9.2.3 Applications of Prioritized Defaults

A third interesting direction is to apply prioritized[1] default reasoning, e.g., in circumscription, and our GNMR architecture, to various areas. **Recall** sub-section 1.6.1, where we outlined various kinds of applications of default / NM reasoning. Next, we list some applications that we find especially interesting.

Inductive Learning:
In inductive learning with background knowledge, prioritized defaults can be used to represent the knowledge base. Building on this thesis (especially the generalization of prioritization and CLD), we have done, elsewhere, some work along this avenue: see "follow-ons" below.

Follow-On Work: Declarative Bias:
In the **Declarative Bias** approach [Grosof and Russell, 1990] (see also [Russell and Grosof, 1987] [Russell and Grosof, 1990b] [Russell and Grosof, 1990a]), inductive bias and current ("working") hypotheses are derived by logical inference from declaratively-represented background knowledge that includes deeper bias. This inference is, in general, non-monotonic: background knowledge takes the form of prioritized defaults, not just for-sure premise beliefs. A very useful and commonly available kind of such knowledge is in the first-order formula form of **determinations**. Updates include empirical instance observations.

Inductive Logic Programming:
Inductive Logic Programming (see the collection in [Muggleton, 1992]), developed independently at about the same time as Declarative Bias, takes a similar approach: to derive inductive hypotheses from declaratively-represented background knowledge, using logic programming techniques. These

[1] or some other similar notion of precedence

ILP derivations are, in general, unsound with respect to first-order logic; that is, they are non-monotonic. A very interesting direction for future research is to apply the tools of NM logical formalisms, e.g., the results of this thesis, to this non-monotonic reasoning. Some early work in this direction is [Bain and Muggleton, 1992]. ILP has attracted a great deal of interest recently in the machine learning research community.

Follow-On Work: Prioritizing Multiple, Contradictory Sources in Learning by Taking Advice:
Another important mode of learning is by taking advice [Dietterich, 1982] (advice in the sense of [McCarthy, 1968]). Important kinds of advice for learning agents are from communication and reading, for example. Multiple sources of advice may offer defeasible advice that conflicts: thus a representational challenge is to treat the precedence associated with sources as itself the subject of reasoning, e.g., to derive precedence from reasoning about reliability or authority. Beliefs about precedence may themselves evolve non-monotonically. In [Grosof, 1992a], we generalize prioritized circumscription to represent such advice-taking in learning agents. Note that this kind of advice-taking is part of the broader process of "knowledge assimilation" (as in the AAAI-92 Spring Symposium of that title).

Plausible Analogy:
Default determinations (see discussion of Declarative Bias above) are also a promising direction for modelling plausible analogy (see discussion of analogy in [Russell, 1989] and [Loui, 1989], for example). However, the flavor of analogical plausible inferences is often probabilistic: see discussion of probabilities below.

The Frame Problem:
The frame problem is the problem of default reasoning about persistence of properties across time or states, e.g., in reasoning about the effects of actions. The frame problem is a well-known problem in NM updating and belief revision (see, e.g., [McCarthy and Hayes, 1969]). We believe the results and approach in this thesis apply to the frame problem.

Other Applications:
There are many other interesting applications of prioritized defaults. Next, we list some that have not been much discussed in the NMR literature.

- Law: Prioritized defaults appear a useful tool to represent the dialectical flavor of arguments and rationales.
- Design Rationales: (for a like reason)
- Expert Systems Rule Interactions: to use prioritized defaults as a tool for compilation, analysis, and verification of rule-based expert systems: here rules may conflict, and the "qualification problem" [Mccarthy, 1977][2] arises.

[2] See also the Introduction of [Ginsberg, 1987] for an explanation of the qualification problem and the frame

Recall also the other applications of NMR, e.g., diagnosis, listed in sub-section 1.6.1.

9.2.4 Improve Expressive Capabilities

A fourth interesting direction is to improve the expressiveness of NMR, including of prioritized defaults.

Bayesian Probabilistic Reasoning:
An important challenge for representation of NMR is to integrate defeasibility with probability (e.g., to represent plausible analogy). Most NM logical formalisms are oriented towards 0-1 (True/False) belief, rather than probabilistic. Yet much of available information is statistical and probabilistic. There has always been interest within the NMR research community in viewing probabilities as an underlying justification for default axioms or NM dynamics of belief, e.g., the recent development by Geffner and Pearl of epsilon-semantics (see, for example, [Geffner, 1992]). But we are equally concerned with another aspect of the connection between probability and NMR: namely, that probabilistic reasoning is *itself* non-monotonic: witness the phenomenon of "reference classes" in the statistical literature (briefly reviewed, for example, in [Loui, 1989]). In [Grosof, 1988], we observed that Bayesian reasoning is itself logically non-monotonic, and that its most basic pattern can be represented in terms of prioritized defaults, somewhat similarly to 0-1 default inheritance. [Bacchus, 1990] takes a similar approach. Bayesian reasoning is important, as well, because it underlies decision-theoretic (i.e., probabilistically-expected utility) reasoning. Much more remains to be done along this avenue.

Follow-On Work: Defeasible Reasoning about Precedence and about What Is Adopted as a Default Rule:
Building on this thesis (especially, the generalization of prioritization, and CLD), we have developed a variant of prioritized circumscription, called Defeasible Axiomatized Policy (DAP) circumscription, that is more expressively powerful than prioritized default circumscription. DAP circumscription enables defeasible reasoning about precedence and about what is adopted as a default rule. **Recall:** our discussion of "Learning by Taking Advice" above, and the discussion of DAP circumscription in section 8.3 and in sub-section 2.11.5. CLD is essentially equivalent to a special case of DAP circumscription. Interestingly, our canonical decomposition results are useful, beyond the default / modules case, for DAP circumscriptions.

Other Expressiveness Issues:
Some other interesting avenues for increasing expressiveness include:

- Higher-Order Base languages in circumscription: [Lifschitz, 1990a] discusses some examples of this.

- Cyclic Precedence: to permit precedence to be a pre-order rather than a strict partial order.

problem.

- Non-Well-Founded Precedence: recall the discussion in sub-section 2.11.5.

Recall also the discussion in sub-section 2.11.5 of expressiveness issues for future work on generalizing prioritization.

9.2.5 Relationships to Other NM Formalisms

A fifth interesting direction is to establish relationships between prioritized circumscription and other NM formalisms, and to apply our approach and results to them.

In sections 8.5 through 8.8, and in section 2.11, we discussed relationships of prioritized circumscription to a variety of other NM formalisms. Some of the most interesting as avenues for investigation are:

- Default Logic and Poole's system (section 8.5)
- Autoepistemic Logic (section 8.6)
- logic programs (section 8.7), including beyond the stratified class
- default inheritance (sub-section 2.4.1)
- model preference logics (sub-section 2.11.2)
- Geffner's [1992] formalism (sub-section 2.11.1)
- Brewka's [1989a] [1989b] formalism (sub-section 2.11.3 and section 8.8)
- Rathmann and Winslett's [1989] variant of circumscription which treats equality differently and avoids nasty unsatisfiability
- argument systems, e.g., those of Loui [1987] and Pollock [1987]
- Ginsberg's [1988] Multi-Valued Logic
- Nebel's [1989] formalism and related work on "epistemic change", e.g., by Gardenfors (see Nebel's paper for review)
- Morgenstern's and Guerriro's [1991] Multi-Agent Non-Monotonic Logic

One interesting goal for such investigations is to combine the expressive and other advantages of these other formalisms with those of circumscription, e.g., explicit non-layered prioritization and skepticality.

9.2.6 Compromise with Resource Constraints

Finally, the most important direction for future research is to find some method for resource-bounded rationality in NMR.

The biggest problem today with expressively rich NMR is the impracticality of its computational cost. One strategy, which we have pursued in this thesis, and discussed earlier in this section, is to guarantee localities for expressively restricted cases, and to exploit those cases when found in an expressively general setting. Another interesting strategy is to approximate while guaranteeing soundness (recall section 8.1).

But we strongly suspect that challenging applications will intrinsically require lower computational cost than can be achieved with these strategies alone. In our view, the biggest research challenge in NMR is to find a **principled approach to trading off soundness, and completeness, for computational resource savings**. This is a problem for much of AI, but is especially important for expressively rich NMR since the computational complexity is especially nasty (recall sub-section 4.2.2).

Decision theory (probabilistically-expected utility[3]) provides a basis for such a principled trade-off. Recently, it has received much attention in other areas of AI. We believe it is very much worth pursuing in NMR as well.

[3] in the sense investigated in mathematical economics; see, for example, [Raiffa, 1968] for an introduction

Appendix A

Introduction to the Proof Appendix

Guide to Reader: READ THIS CHAPTER BEFORE READING ANY OF THE PROOFS IN THE APPENDIX!!!

Terminology: We will refer to chapters 1–9 as: **"the main text"** or "the main body".

In the main text, we gave many of the shorter proofs. For the rest of the results, however, we gave only a proof overview in the main text, and/or referred you to this Appendix for the full proof.

In support of these proofs, this Appendix includes notation, terminology, definitions, facts, propositions, and lemmas that are not included, or even mentioned, in the main body of the text.

Our reasons for making a separate Appendix are to improve the main text's flow, readability, and accessibility.

A.1 Organization and Overview

The organization of the proof Appendix generally parallels that of the main text. In addition, however, there are some sections of background material that are "broken out" separately from the flow, for ease of reference.

*****: Basic notation, terminology, and definitions are in sections A.2, B.1, and B.5: these are very important to read to be able to understand the following parts of the proof Appendix.**

Each chapter in the Appendix after this, the first one, corresponds to a chapter from 2 to 6 in the main text. Each Appendix section whose title begins with "For ..." includes proofs for a section or sub-section of the main text. Within each of these Appendix sections are definitions, lemmas, etc., used for the proof of the main-text results (typically, Theorems) in that section. These precede the proofs of the main-text results. For convenience when reading, all statements of main-text results (typically, Theorem statements) are repeated in the Appendix.

Overview of Dependencies among Results

As a general rule, results build sequentially: earlier results do not rely on any results that come later in either the sequence of the main text or the sequence of the appendix. There are a very few exceptions to this sequentiality: these are explicitly noted, and do not cause any circularity.

Theorem 2.19 (fixture in the minimized pre-order) and Theorem 2.34 (well-definition of prioritization) are so basic that they are, in effect, built into the notation and concepts that follow them.

Theorem 2.58 (the monotonicity of prioritization) and Lemma B.10 (prioritization does not influence "tying") are used very often.

The proofs of Theorems 5.14, 5.32, and 5.35, about canonical conjunctive and serial decompositions along prioritization (from sections 5.4 and 5.5), are quite complicated: we develop many lemmas and definitions to support them.

These general-case canonical decomposition results are used in the proofs of most of our later, special-case decompositions and safeties.

A.2 Notation, Terminology, Style Conventions

Terminology: In this Appendix, **"Local"** means local to the current (part of a) proof, unless some other meaning is explicitly stated, or unless it is clear from context that we mean local in the sense of localities of NM inference etc..

Line Numbering Notation: An "A" prefix on a line number, e.g., in "(A1)", indicates an Assumption, or premise, or a step working forward from such. A "G" prefix, by contrast, indicates a Goal, or a step working backward from such. A "D" prefix stands for a Definition. A "S" prefix indicates a Sub-proof: sometimes we nest these. Line numbering is local to a proof or section of a proof or sub-proof etc.: when in doubt, apply a specificity principle to disambiguate.

Proof Step Notation: " $\Longrightarrow$ " indicates an implication at the (meta-linguistic) level of the proof. Similarly, "$\overset{\text{def}}{\Longleftrightarrow}$" and "$\Longleftrightarrow$" indicate equivalences at the level of the proof.

Superscript Notation: A set as a superscript will usually indicate either a tuple, or a conjunction over the members of the set.

Notation: Parentheses for Formulas: In the main text, we usually distinguished formulas from atoms by using square brackets instead of parentheses: e.g., $B[Z]$ or $D[Z, x]$ versus $ab1(x)$ or $Pj(xj)$. In this Appendix, we will relax this convention and often use parentheses in formulas, e.g., to write $B(Z)$ or $D(Z, x)$.

Such-That Notation "$\ni$" stands for "such that".

Terminology: -Free We say that a formula G is *Q-free* when the symbols Q do not appear in G.

Appendix B

Proofs for Chapter 2

B.1 Notation for Pre-Orders

Guide to Reader: Pre-requisite: Definition 2.1. **THIS SECTION IS VERY IMPORTANT TO READ TO BE ABLE TO UNDERSTAND THE REMAINDER OF THE PROOF APPENDIX.**

Terminology: Strict Version of Pre-Order:
We call $\prec_H$ the *strict version* of the pre-order $\preceq_H$.

Notational Convention for Pre-Orders: Omitting Arguments:
For brevity, in expressions involving several appearances of pre-orders and/or their strict versions, we will often omit their arguments, when there is a common "left" argument and a common "right" argument throughout the expression.

Suppose the implicit arguments are Z "on the left" and Z' "on the right". Some examples of this convention are:

$\preceq_G \wedge \preceq_H$ stands for $(Z \preceq_G Z') \;\wedge\; (Z \preceq_H Z')$

$\preceq_G \vee \prec_H$ stands for $(Z \preceq_G Z') \;\wedge\; (Z \prec_H Z')$

$\neg \preceq_G$ stands for $\neg (Z \preceq_G)$

$\approx_G \supset \neg \prec_H$ stands for $(Z \approx_G Z') \;\supset\; \neg (Z \prec_H Z')$

$\bigwedge i \in S \approx_{Gi} \;\supset\; (\neg \succeq_{Hi} \wedge \succ_{Li})$ stands for $\bigwedge I \in S (Z \approx_{Z}') \;\supset\; [\neg (Z \succeq_{Hi} Z') \wedge (Z \succ_{Li} Z')]$

We will also use this convention when constructing relations. E.g., we will sometimes use

$\preceq_G \wedge \preceq_H$ to stand for $\lambda Z, Z'. \, (Z \preceq_G Z') \;\wedge\; (Z \preceq_H Z')$

B.2 Properties of Pre-Orders

Guide to Reader: Pre-requisite: Definition 2.1.

Fact B.1 (Strict Version of Pre-Order Is A Strict Partial Order)

When $\preceq_H$ is a pre-order, $\prec_H$ is strict partial order; that is, $\prec_H$ is anti-reflexive and transitive.

Fact B.2 ("Non-Uniqueness" of Strict Versions)
For any strict partial order $\prec_H$, there is a class of pre-orders such that $\prec_H$ is their strict version. Consider

$$\preceq_L \stackrel{\text{def}}{\equiv} \prec_H \vee \approx_F$$

where $\preceq_F$ is any pre-order such that

$$\approx_F \supset \neg\prec_H$$

Then $\prec_L \equiv \prec_H$. If $\approx_F \equiv \approx_H$, then $\preceq_L \equiv \preceq_H$. However, $\approx_F$ could be the identity relation $\lambda Z, Z'.\ Z{=}Z'$. For most pre-orders $\preceq_H$ of interest, $\approx_H$ is *not* equivalent to the identity relation. Thus, in general, and usually for our pre-orders of interest, the mapping from strict versions to pre-orders is not unique. Note that, for any strict partial order, there is at least one pre-order for which it is the strict version. Simply choose $\approx_F$ to be the identity relation.

Fact B.3 (Pre-Order is a Partial Order Over Equivalence Classes)
Every pre-order $\preceq_H$ defined over domain $\mathcal{D}$, is isomorphic to a partial order defined over the domain $\mathcal{EH}$, where $\mathcal{EH}$ is the set of all equivalence classes under $\approx_H$ over $\mathcal{D}$.

B.3 For sub-section 2.3.1: Fixture in the Minimized Pre-order

Lemma B.4
Let

$$\preceq_L \stackrel{\text{def}}{\equiv} (\preceq_G \wedge \preceq_H)$$

Then

$$\prec_L \equiv [(\prec_G \wedge \preceq_H) \vee (\prec_H \wedge \preceq_G)]$$

Proof : Straightforward manipulation using definition of $\prec$ as $\preceq \wedge \neg\succeq$. **QED**

Theorem 2.19 (Expressing Fixture in the Minimized Pre-Order)
Circumscription with fixture of an equivalence relation can be reduced to general circumscription: one can express the fixture as part of the minimized pre-order. Let $\preceq_L$ be defined as $\preceq_H \wedge \approx_F$. Then $\preceq_L$ is a pre-order, and

$$C(B;L;Z) \equiv C(B;H;fix\ F;Z)$$

Proof Overview: Fairly straightforward. See Appendix. □

Proof of Theorem 2.19 : $C(B;L;Z) \stackrel{\text{def}}{\Longleftrightarrow}$

$$B(Z) \wedge \neg\exists Z'.\ B(Z') \wedge Z' \prec_L Z \qquad (1)$$

Since $\approx_F$ is a pre-order, let us write it as $\preceq_G$. Then

$$\prec_G \equiv False \tag{2}$$

By the definition of $\preceq_L$ and Lemma B.4,

$$\prec_L \;\equiv\; [(\prec_H \wedge \preceq_G) \wedge (\preceq_H \wedge \prec_G)] \tag{3}$$

which, by (2), $\implies$

$$\prec_L \;\equiv\; (\prec_H \wedge \preceq_G) \tag{4}$$

(4) $\implies$ (1) is equivalent to

$$B(Z) \wedge \neg\exists Z'.\, B(Z') \wedge Z' \prec_H Z \wedge Z' \approx_F Z \tag{5}$$

which is just the definition of $C(B; H; fix\ F; Z)$.

QED Theorem 2.19

B.4 For sub-section 2.3.2: Fixing of Functions

Theorem 2.24 (Fixing of Functions)
Some equivalent and sufficient conditions for the fixing of functions are as follows. The first two points are new as far as we are aware. The last two follow easily from the second.

1. Fixing the function f is equivalent to fixing the elementary formula
   $\lambda x, y.\ f(x) = y$.
   Thus, one can represent fixing functions as fixing elementary formulas.

2. Suppose the base B of a circumscription $C(B; H; Z)$ includes a complete theory of equality. Then all function symbols are effectively fixed, i.e., **indirectly** fixed (**terminology**; see Definition 3.47). More precisely: let $F \subset Z$ stand for the tuple of all function symbols. Then
   $C(B; H; fix\ F; Z) \;\equiv\; C(B; H; Z)$

3. Domain closure plus a complete theory of equality about the ground terms suffices to indirectly fix (in the sense above) all the functions.

4. Domain closure plus uniqueness of names suffices to indirectly fix (in the sense above) all the functions.

Proof Overview: We show each point from the previous one. See Appendix. □

Proof of Theorem 2.24 : We emply the notion of fixed-relative and several of its properties: see the beginning of sub-section 3.5.5, especially, Definition 3.47 and Proposition 3.48, which we use below. (See also Definition 3.49, Definition 3.50, and Lemma 3.51, as background.)

(1.) Let Z be the tuple of all predicate and function symbols in the base language $\mathcal{L}$. Let the pre-order H be defined over Z. Let the function f be a member of Z. The definition (Definition 2.14) of fixing f, within the overall pre-order H, is that: $Z \preceq_H Z'$ implies

$$\forall x.\ f(x) = f'(x) \tag{1}$$

Here x ranges over object tuples of arity m, where m is the arity of f.

The definition of fixing the formula $f(x) = y$, within the overall pre-order H, is that: $Z \preceq_H Z'$ implies

$$\forall x, y.\ (f(x) = y) \equiv (f'(x) = y) \tag{2}$$

It suffices to show that (1) is equivalent to (2); i.e., that the pre-order for fixing a function symbol f is equivalent to the pre-order for fixing the open formula $f(x) = y$.

That (1) implies (2) follows from the tautology:

$$\forall x, y.\ f(x) = f'(x) \quad \supset \quad \{[f(x) = y) \supset f'(x) = y)] \ \wedge \ [f'(x) = y \supset f(x) = y)]\}$$

That (2) implies (1) follows from the tautology:

$$\forall x.\ \{[\forall y.\ f(x) = y \supset f'(x) = y] \quad \supset \quad f(x) = f'(x)\}$$

To see this, substitute $f(x)$ for y.

QED (1.)

(2.) Examine the definition of a complete theory of equality: Definition 2.22. By Propositions 3.48 and 3.35, this implies that each formula of the form $x = y$ is fixed relative to the circumscription. Therefore, in particular, each formula of the form $f(x) = y$, where $f \in F$, is fixed relative to the circumscription. Therefore, by part **(1.)** of the Theorem, each function $f \in F$ is fixed relative to the circumscription.

QED (2.)

The conditions in **(3.)** and **(4.)** are just special cases of complete theory of equality (**(2.)**).

QED Theorem 2.24

B.5 Notation, Terminology, and Definitions for Prioritized Pre-Orders

Guide to Reader: Pre-requisite: Definition 2.33.

THIS SECTION IS VERY IMPORTANT TO READ TO BE ABLE TO UNDERSTAND THE REMAINDER OF THE PROOF APPENDIX.

Terminology: Dominate:

When $R(j, i)$ holds, we say that j *dominates* i, and that j is a *dominator of* i. We call $R_D(i)$ the dominators of i, or the *dominator set* of i.

Definition 5.13 (Prioritization Partitions)

(We find it convenient to introduce this definition earlier in the Appendix's sequence of proofs than we do in the main text.)

Let the index tuple N be the domain of a prioritization partial order R. For example, N indexes a tuple of starting pre-orders H, say, a tuple P of minimized predicates. Relative to that prioritization p.o. R, we define, for each index $i \in N$, the partition of the remainder of N into the indices that are strictly higher priority than (Dominate) i, those that are strictly lower priority than (Dominated By) i, and those that are neither higher nor lower priority than (Incomparable to) i.

$$R_D(i) \stackrel{\text{def}}{=} \{j|R(j,i)\}$$
$$R_{DB}(i) \stackrel{\text{def}}{=} \{j|R(i,j)\}$$
$$R_I(i) \stackrel{\text{def}}{=} \{j|\neg R(i,j) \wedge \neg R(j,i) \wedge j \neq i\}$$

Identity: $\quad \forall i. \quad N = \{i\} \cup R_D(i) \cup R_I(i) \cup R_{DB}(i)$

Domain of R may be beyond N:
R may be defined over a superset of the index set N. We can write

$$\preceq_{R^N} \stackrel{\text{def}}{\equiv} \forall i \in N. \approx_{D(i)\cap N} \supset \preceq_i$$

However, as an implicit convention, we take R to be defined only over N, unless, in context, it is clearly otherwise.

Notation and Terminology:
We will sometimes make the index tuple N appear explicitly, i.e., to write H as H^N, and R as R^N. Relative to the prioritization operation, we will refer to H^N as the *starting* pre-orders. More generally, we write H^E to stand for the subtuple of H^N, and R^E to stand for the projection (restriction) of R^N, defined by the index set (tuple) $E \subseteq N$.

Abbreviations in Notation:
When discussing abstract properties of prioritized pre-orders, we will often leave their arguments (e.g., Z and Z'), their starting pre-order tuples (e.g., H^N), and even their index sets (e.g., N) implicit. For example, for $Z' \preceq_{H^N;R} Z$, we will often write simply $\preceq_{R^N}$, or even $\preceq_R$. For $\preceq_{Hi}$, we will often write simply $\preceq_i$.

As a convention, if we do not mention them explicitly: N stands for the index set; H^N refers to the tuple of starting pre-orders, and is defined over Z; and R is the prioritization partial order and is defined over N (and perhaps, over a superset of N).

We will also often abbreviate our notation for prioritization partitions, so as to leave the prioritization R, and often the index set N, implicit. We will often write $D(i)$ for $R_D(i)$, $I(i)$ for $R_I(i)$, and $DB(i)$ for $R_{DB}(i)$. In addition, we will write $DI(i)$ for $D(i) \cup I(i)$, and $IDB(i)$ for $I(i) \cup DB(i)$.

We will often write i to stand for the singleton index set $\{i\}$.

For $\forall i \in N. \ \ldots$, we will often write $\forall i. \ \ldots$; likewise for $\exists i \in N. \ \ldots$, we will often write $\exists i. \ \ldots$. For $(\forall i \in E. \approx_i)$, we will often write $\approx_E$.

Sometimes, for emphasis, or to help reduce the chance of confusion, we will make something, e.g., N, explicit in only part of an expression.

Notation: Prioritized Default and Predicate Pre-Orders:
We will sometimes write $Z \preceq_{P;R} Z'$, where the predicate tuple P is a sub-tuple of Z, as: $P \preceq_R P'$. Likewise, we will sometimes write $Z \preceq_{D;R} Z'$, where D is a tuple of default formulas, as: $D[Z] \preceq_R D[Z']$.

Identity: Note that $P \preceq_{\emptyset} P'$ is equivalent simply to $P \leq P'$. Likewise, $D[Z] \preceq_{\emptyset} D[Z']$ is equivalent simply to $D[Z] \leq D[Z']$.

Definition B.5 (Projection of a Prioritized Pre-Order)
We find it useful to define the projection of a prioritized pre-order, defined with index set N, onto any subset of those indices.
Let $E \subseteq N$. We define the *projection* of $\preceq_{R^N}$ onto E, as:

$$\preceq_{R^E} \stackrel{\text{def}}{\equiv} \forall i \in E.\ \approx_{D(i) \cap E} \supset \preceq_i$$

Definition B.6 (Selected Segment of a Prioritized Pre-Order)
Also, we find it useful to define another sort of part of a prioritized pre-order, again corresponding to a subset of the indices.
Let $E \subseteq N$. We define the *segment* of $\preceq_{R^N}$ *selected* by E, as:

$$\preceq_{R^N:E} \stackrel{\text{def}}{\equiv} \forall i \in E.\ \approx_{D(i) \cap N} \supset \preceq_i$$

which, if we leave N implicit, is:

$$\preceq_{R^N:E} \stackrel{\text{def}}{\equiv} \forall i \in E.\ \approx_{D(i)} \supset \preceq_i$$

Warning: LaTeX Glitch:
Due to a LaTeX glitch, in the immediately following kind of expressions, you will often see spurious variation in the sizes of "R" and index sets such as "N". Please ignore the sizing and just pay attention to subscript and superscript relationships.

Identities: Notice that

$$\preceq_{R^N} \equiv \preceq_{R^N:N}$$

$$\preceq_{R^N} \equiv \forall i \in N.\ \preceq_{R^N:\{i\}}$$

$$\forall E \subseteq N.\ \preceq_{R^N} \supset \preceq_{R^N:E}$$

$$\preceq_{R^N:(E \cup F)} \equiv \preceq_{R^N:E} \wedge \preceq_{R^N:F}$$

Definition B.7 (Upwardly-Closed with respect to Priority)
Let R be a prioritization defined over N. We say that the index subset $E \subseteq N$ is *upwardly-closed*, or *up-closed*, with respect to priority, relative to the prioritization R, when all dominators of the indices in E lie in E:

$$\forall i \in E.\ R_D(i) \subseteq E\ .$$

Or more concisely,

$$\forall i \in E.\ D(i) \subseteq E$$

Definition B.8 (Refinement of a Strict Partial Order)
Let S and R each be a strict partial order (e.g., a prioritization) defined over N. We say that R *refines (extends)* S, or, equivalently, that R is a *refinement (extension)* of S, when

$$\forall x, y.\ S(x,y) \supset R(x,y) \tag{0}$$

holds. We can write (0) more concisely: as $S \leq R$.

Viewing the strict partial orders as directed acyclic graphs, condition (0) says that R includes every arc in S, plus perhaps some additional arcs. Intuitively, when S and R represent prioritization p.o.'s, R then specifies at least as much prioritization dominance relationships as does S. Refining a prioritization partial order thus means "adding" prioritization, as in a prioritization update in CLD (CLD is defined in sub-section 3.2.1).

Well-Founded Induction:
Well-founded induction, a standard mathematical notion, is a generalized form of mathematical induction that permits any well-founded strict partial order, rather than only the natural numbers, to play the role of an indexing "sequence". Reviewing it briefly: one assumes that the (parametrized) goal holds for all predecessors (with respect to the partial order) of a given, arbitrary index. Then one tries to show that the goal follows for that index. If successful, it follows by induction that the goal holds for all indices (in the domain of the partial order).

Many of our proofs will employ well-founded induction, typically on the prioritization partial order R.

To see why well-founded induction is justified: consider the following sketch of a proof by contradiction.[1] Suppose the principle of well-founded induction is false, for some paramaterized goal proposition $G(i)$, defined over indices i in the domain N, with respect to a well-founded strict partial order R. Consider the set V of all indices for which the goal proposition does not hold:
$V \stackrel{\text{def}}{=} \{j \in N | \neg G(j)\}$
Since R is well-founded over N, R must thus be well-founded over (i.e., when restricted to) the set V which is a subset of N. Therefore, this set V must have at least one minimal element k with respect to the ordering R. Then all of k's predecessors (in N) with respect to R satisfy G, but k does not. $\Rightarrow\Leftarrow$

For a detailed treatment of mathematical induction, including over well-founded orderings, see [Quine, 1969].

B.6 For section 2.7: Well- and Ill- Definition of Prioritization

Theorem 2.34 (Well-Definition)
Prioritization according to Definition 2.33 is well-defined for well-founded prioritization partial orders.

[1] Thanks to Francisco Corella (private communication) for this proof sketch, and for introducing us to well-founded induction.

Let R be a well-founded prioritization defined over N.[2] Let H be a tuple of pre-orders, indexed by N, each defined over a common domain. Then $(H;R)$ is a pre-order, defined over the same domain.

Proof Overview: Quite non-trivial. Reflexivity of $(H;R)$ is easy to show. However, our proof of the transitivity of $(H;R)$ is more complicated: it is inductive, using the well-foundedness of R. See Appendix for full proof. □

Proof of Theorem 2.34 :

We want to show that $(H;R)$ is I) reflexive and II) transitive. We are given that each Hi, for all $i \in N$, is reflexive and transitive.

I) $(H;R)$ is reflexive:

$$\begin{aligned}
&\stackrel{\text{def}}{\equiv} \ \forall Z.\, Z \preceq_L Z \\
&\stackrel{\text{def}}{\equiv} \ \forall Z.\, \forall i \in N.\, [\forall j \in N.\, R(j,i) \supset (Z \approx_{Hj} Z)] \ \supset \ (Z \preceq_{Hi} Z)
\end{aligned}$$

For each Z and i, the right hand side follows from the reflexivity of Hi. **QED I)**

II) Transitivity of $(H;R)$

$$\stackrel{\text{def}}{\Longleftrightarrow} \ \forall A,B,C.\ A \preceq_{(H;R)} B \wedge B \preceq_{(H;R)} C \supset A \preceq_{(H;R)} C$$

$$\Longleftrightarrow \ \forall A,B,C.$$

$$\forall i.\, [\forall j.\, R(j,i) \supset A \approx_{Hj} B] \supset A \preceq_{Hi} B \tag{A1}$$

$$\wedge\, \forall i.\, [\forall j.\, R(j,i) \supset B \approx_{Hj} C] \supset B \preceq_{Hi} C \tag{A2}$$

$$\supset$$

$$\forall i.\, [\forall j.\, R(j,i) \supset A \approx_{Hj} C] \supset A \preceq_{Hi} C \tag{G1}$$

Assume (A1), (A2); it suffices to show (G1).

The quantifiers for indices (e.g., i and j) are understood to range over N, in the rest of this proof.

Note that (A1), (A2) $\Longrightarrow$

$$\forall i.\, [\forall j.\, R(j,i) \supset A \approx_{Hj} B \approx_{Hj} C] \ \supset \ (A \preceq_{Hi} B \preceq_{Hi} C) \tag{A3}$$

We introduce the following notation:

$$\begin{aligned}
D(i) &\stackrel{\text{def}}{=} \{j \mid R(j,i)\} \\
TE^j &\stackrel{\text{def}}{\equiv} A \approx_{Hj} B \approx_{Hj} C \\
OE^j &\stackrel{\text{def}}{\equiv} A \approx_{Hj} C \\
TL^j &\stackrel{\text{def}}{\equiv} A \preceq_{Hj} B \preceq_{Hj} C \\
OL^j &\stackrel{\text{def}}{\equiv} A \preceq_{Hj} C
\end{aligned}$$

This is intended to be mnemonic: "D" is an acroynym for "Dominator set of"; "TE" is an acronym for "Two Equal", "OL" for "One Less-than-or-equal", etc..

[2] Actually, R could be defined beyond N. More generally and precisely, then, let the restriction of R to N be well-founded.

We permit sets, not just singletons, to appear as superscripts. This stands for conjunctions. E.g.:

$$TE^{D(i)} \stackrel{\text{def}}{\equiv} \forall j \in D(i).\, TE^j$$
$$OE^{D(k)} \stackrel{\text{def}}{\equiv} \forall j \in D(k).\, OE^j$$

(Recall that we introduced this notational convention in section B.5.)

Re-stating: it suffices to show that

$$\forall i.\, TE^{D(i)} \supset TL^i \tag{A3}$$

implies

$$\forall i.\, OE^{D(i)} \supset OL^i \tag{G1}$$

By the transitivity of each Hj:

$$\forall j.\, TL^j \supset OL^j$$

Thus, comparing (A3) and (G1), it suffices to show

$$\forall i.\, OE^{D(i)} \supset TE^{D(i)} \tag{G2}$$

We prove (G2) by well-founded induction on the partial order R.
Let i be an arbitrary index (in N). Our inductive assumption is

$$\forall k \in D(i).\, OE^{D(k)} \supset TE^{D(k)} \tag{A4}$$

We also assume the left-hand side of (G2):

$$OE^{D(i)} \tag{A5}$$

Then it suffices to show

$$TE^{D(i)} \tag{G3}$$

We use the following local lemma:

$$\forall j.\, TL^j \supset (OE^j \supset TE^j) \tag{Local Lemma 1}$$

i.e.:

$$(A \preceq_{Hj} B \preceq_{Hj} C) \supset [(A \approx_{Hj} C) \supset (A \approx_{Hj} B \approx_{Hj} C)]$$

Proof of Local Lemma 1: By contradiction.
Assume $(A \preceq_{Hj} B \preceq_{Hj} C)$ and $\neg(A \approx_{Hj} B \approx_{Hj} C)$. It follows that either $(A \prec_{Hj} B)$ or $(B \prec_{Hj} C)$.
But then $(A \preceq_{Hj} B \preceq_{Hj} C)$ and the transitivity of Hj imply $A \prec_{Hj} C$ which contradicts $A \approx_{Hj} C$.
QED Local Lemma 1

Comparing (A5) and (G3), using (Local Lemma 1), $\Longrightarrow$ it suffices to show :

$TL^{D(i)}$ (G4)

By the transitivity of R, we have that fact that:

$\forall k \in D(i).\, D(k) \subseteq D(i)$ (A6)

(A6), (A5) $\implies$

$\forall k \in D(i).\, OE^{D(k)}$ (A7)

(A7), (A4) $\implies$

$\forall k \in D(i).\, TE^{D(k)}$ (A8)

(A8), (A3) $\implies$

$\forall k \in D(i).\, TL^{k}$ (G4)

QED II

QED Theorem 2.34

Theorem 2.37 (Ill-Definition)
Prioritization according to Definition 2.33 fails to be well-defined when the prioritization partial order is not well-founded.

For any non-well-founded prioritization, there is a starting set of pre-orders for which the result of prioritization according to Definition 2.33 fails to be transitive.

Proof Overview: See Appendix for the proof by counterexample. □

Proof of Theorem 2.37 :

Our proof is by counterexample. First we show that transitivity fails for a particular non-well-founded prioritization partial order. Then we generalize.

The fundamental intuition behind the counterexample is that Definition 2.33 leads to a "tie condition" $\approx_{(H;R)}$ that is too weak, i.e., too permissive, when the prioritization partial order is not well-founded.

Let N be the integers (negative as well as positive), representing a discrete timeline, and let each Hi be defined as $\lambda P, P'.\, P(i) \supset P'(i)$, for some propositional fluent P defined on times $i \in N$. Let R be the total order on the timeline, representing temporal precedence, i.e., $R(i,j) \stackrel{\text{def}}{\equiv} (i < j)$, where here $<$ is ordinary arithmetic less-than. Note that R is not well-founded: it contains an infinite chain in the direction of ascending priority (negative time).

Then the relation $L \stackrel{\text{def}}{\equiv} (H;R)$ is not transitive, as is shown by the following three models of P:

Model	$\ldots$	-4	-3	-2	-1	0	$\ldots$
$M1:$	$\ldots$	T	F	T	F	T	$\ldots$
$M2:$	$\ldots$	F	T	F	T	F	$\ldots$
$M3:$	$\ldots$	T	F	T	F	F	$\ldots$

Here T and F stand for $True$ and $False$. Before time 0, all three models "tick-tock" back to negative infinity: at each time point $i < 0$, $P(i-1) \equiv \neg P(i)$. Models $M1$ and $M3$ are "in synchrony"; but both are "out of synchrony" with $M2$. At time 0, however, $M1$ and $M3$ differ. Now consider how the relation L compares the three models. $M1 \sqsubseteq_L M2$.[3] Why? The key observation is that: if at no index i is the following subformula in the definition of $\preceq_L$ satisfied, then the overall pre-order condition $\preceq_L$ is satisfied:

$$\forall j \in N.\ R(j,i) \supset (P(j) \equiv P'(j))$$

Similarly, $M2 \sqsubseteq_L M3$. But $M1 \sqsupset_L M3$, since $M1$ and $M3$ are identical before time 0, and at time $i = 0$, $M1 \sqsupset_{Hi} M3$. Thus transitivity is violated.

This construction of a counterexample extends to *any* non-well-founded prioritization partial order R. As above, let P be defined on indices $i \in N$, where N is the domain of R. Let the starting pre-orders Hi also be defined as above. By the definition of well-foundedness, R must contain an infinite ascending (with respect to priority) chain of indices. Then, let $M1$,$M2$,$M3$ be as above on one such infinite ascending chain, and be extended to be identical to each other everywhere else. Transitivity fails just as it did above. Thus for any non-well-founded prioritization, there is a starting set of pre-orders for which the result of prioritization according to definition 2.33 fails to be transitive.

QED Theorem 2.37

B.7 For sub-section 2.7.1: Default Instances

Theorem 2.47 (Open Default As Parallel Default Instances)
In any prioritized default circumscription (PDC):
Suppose that the base sentence entails domain closure. Suppose also that all functions are fixed or, more generally, fixed "relative" to the circumscription (see Definition 3.47). Then every open default is equivalent to the parallel prioritization of the tuple of all its default instances. I.e., in the context of the circumscription, every (open default's) default pre-order is equivalent to the conjunction of the default pre-orders for that default's instances.

More explicitly: for each $i \in N$,

$$\forall xi.\ Di[Z, xi] \supset Di[Z', xi] \quad \equiv$$
$$(Di[Z, t1] \supset Di[Z', t1]) \ \wedge\ \ldots\ \wedge\ (Di[Z, tm] \supset Di[Z', tm])$$

where $t1, \ldots, tm$ are all of the tuples, of the domain elements, of the appropriate arity for Di.

Note that uniqueness of names, in the presence of domain closure, suffices to effectively fix (i.e., fix "relative") all the functions, by Theorem 2.24.

Note also that the sufficient condition above can be weakened.

See Lemma 3.23 for the generalization of this theorem to the "quasi-propositional" case.

[3]For the definition of the $\sqsubseteq$ notation, see the Preference on Models subsection in section 4.

Proof Overview: Not terribly complicated. See Appendix. □

Proof of Theorem 2.47 : Because the functions are fixed relative to the circumscription, the PDC can be represented as:

$$PDC(B; D^N; R; fix\ F; P, F) \tag{1a}$$

Here, P is the tuple of all predicate symbols in $\mathcal{L}$, F is the tuple of all function symbols in $\mathcal{L}$. Let $Z \stackrel{\text{def}}{=} \langle P, F \rangle$.

Since the function symbols are fixed, they can be omitted from the second-order quantification in the circumscriptive augmentation. Thus, (1a) $\iff$

$$B[P, F] \ \wedge\ \neg\exists P'.\ B[P', F] \ \wedge\ \langle P, F \rangle \prec_{D^N;R} \langle P', F \rangle \tag{1b}$$

The given domain closure condition is:

$$B[P, F] \ \models\ \bigvee_{i=1}^{g} ti[F] \tag{2}$$

Here, each $ti[F]$ is a ground term.

We want to show that (1b) is equivalent to:

$$B[P, F] \ \wedge\ \neg\exists P'.\ B[P', F] \ \wedge\ \langle P, F \rangle \prec_{E^N;R} \langle P', F \rangle \tag{3}$$

where each Ej, for $j \in N$, is the pre-order that results from the parallel prioritization of all of the default instances of the open default $Dj[Z, xj]$:

$$Z \preceq_{Ej} Z' \ \stackrel{\text{def}}{\equiv}\ \bigwedge_{i=1}^{mj} Dj[Z, uji] \supset Dj[Z', uji]$$

Here each uji is a tuple, of ground terms, that has the same arity as xj has.

Assume

$$B[P, F] \ \wedge\ B[P', F] \tag{A}$$

Then it suffices to show that:

$$\langle P, F \rangle \prec_{D^N;R} \langle P', F \rangle \ \equiv\ \langle P, F \rangle \prec_{E^N;R} \langle P', F \rangle \tag{G1}$$

Thus, it suffices to show , for each $j \in N$, that both:

$$\langle P, F \rangle \preceq_{Dj} \langle P', F \rangle \ \equiv\ \langle P, F \rangle \preceq_{Ej} \langle P', F \rangle \tag{G2}$$

and

$$\langle P, F \rangle \succeq_{Dj} \langle P', F \rangle \ \equiv\ \langle P, F \rangle \succeq_{Ej} \langle P', F \rangle \tag{G3}$$

For each $j \in N$: (2), together with the left-hand-side of (A), $\implies$

$$\bigvee_{i=1}^{mj} xj = uji[F] \tag{4}$$

Here, we have shown explicitly that uji mentions function symbols but not predicate symbols.

(4) $\implies$

$$(\forall xj.\ Dj[P, F, xj] \supset Dj[P', F, xj]) \ \equiv$$
$$(\bigwedge_{i=1}^{mj} Dj[P, F, uji[F]] \supset Dj[P', F, uji[F]])$$

which is just the elaboration of (G2). In like fashion, (4) $\implies$ (G3).

QED Theorem 2.47

B.8 For sub-section 2.7.3: Composing Prioritization

Theorem 2.52 (Closure of Well-Founded Class)
The well-founded class of strict partial orders is closed under composition. (For 1 up to any finite number of composition steps.)

Proof Overview: See Appendix. □

Proof of Theorem 2.52 : **One composition step:** By contradiction. Assume there exists an infinite decreasing [4] chain. The first point to notice is that each arc in the chain must be one of two (possibly overlapping) kinds:
1) an arc between a member (say a) of NTi, and a member (say b) of another NTj, where i and j are distinct, and where the arc definitionally arises from $R1$, i.e., $R1(i, j)$; *or*
2) an arc between two members (say a and b) of the same NTi, where the arc definitionally arises from RTi, i.e., $RTi(a, b)$.
By "definitionally arises" here, we mean arising from (one of) the two disjunctive possibilities in Definition 2.51.

Since $R1$ is well-founded, there can only be a finite number of type-1 arcs in the chain. Thus there must be an infinite number of type-2 arcs. But no particular RTi can contribute more than a finite number of type-2 arcs, since each RTi is also well-founded. And for there otherwise to be an infinite number of type-2 arcs in the chain requires an infinite number of interleaving type-1 arcs. $\Rightarrow\Leftarrow$

Finite number of composition steps: By induction: use the one-step case of Theorem.

QED Theorem 2.52

Definition B.9 (Prioritization Depth)
Let R^N be a well-founded prioritization partial order. We define the *depth* of an index $i \in N$ in the prioritization R^N as the length of the shortest chain in R^N that ends in i, and begins at an index with no dominators in N. Thus an index with no dominators in N has depth 0. Notice that no index in N can have a depth greater than the size of N. One can view the prioritization partial order as a directed acyclic graph (dag), where each node represents an index, and each arc represents an immediate-neighbor relationship. Then a source (a node with no incoming arcs) of the dag corresponds to an index with no dominators, and depth of an index i is the minimum number of arcs required for a path from a source to the node corresponding to i. Note that every index has a unique depth. Thus N can be partitioned according to depth. Notice also that every dominator of i has strictly smaller depth than i.

Lemma B.10
Let R^N be well-founded. (Note that N may be infinite.) Then

[4] In the usual mathematical convention for partial orders, $S(x, y)$ means that x is less than y in the p.o. S. Remember, however, that for prioritization partial orders, we reverse this convention, and treat $S(x, y)$ as meaning x has *higher* priority than y. In this proof, we employ the usual convention.

$$\approx_{R^N} \quad \equiv \quad \forall i \in N. \approx_i \tag{G}$$

I.e., more concisely:

$$\approx_{R^N} \quad \equiv \quad \approx_N$$

Proof of Lemma B.10 : From its definition, $\approx_{R^N} \iff$

$$\forall i \in N. \approx_{D(i)} \supset \approx_i \tag{1}$$

Thus the right-to-left direction of the lemma is easy. To show the left-to-right direction: assume the left-hand-side of (G). We show the right-hand-side of (G) by induction on the depth of i in the prioritization partial order.
Let $N[d]$ stand for $\{i \in N \mid depth(i) = d\}$, for $d \geq 0$.
Let $N[0:d]$ stand for $\{i \in N \mid depth(i) \leq d\}$. Our inductive hypothesis is

$$\forall i \in N[0:d]. \approx_i \tag{2}$$

I.e., $\approx_{N[0:d]}$. Our inductive index is d.
Base case of the induction: $d = 0$. Since

$$\forall i \in N[0]. D(i) = \emptyset$$

, (1) thus implies (since $N[0] = N[0:0]$) that

$$\forall i \in N[0:0]. \approx_i$$

Inductive case: Assume the case for depth d, i.e., (2). Since every dominator of i has strictly less depth than i,

$$\forall i \in N[d+1]. D(i) \subseteq N[0:d]$$

$\implies$

$$\forall i \in N[d+1]. \quad \approx_{N[0:d]} \supset \approx_{D(i)} \tag{3}$$

(1),(2),(3) $\implies$

$$\forall i \in N[d+1]. \approx_i$$

QED Lemma B.10

Generalization of Lemma B.10:
Note that the proof shows that this lemma generalizes: any up-closed S that is a subset of N may play the role of N. That is:

$$\approx_{R^S} \quad \equiv \quad \approx_S$$

Theorem 2.56 (Composability of Prioritizations)
Prioritization is composable in the following senses. Firstly, prioritizing pre-orders which are themselves the result of prioritization, is equivalent to prioritizing using the composition of the prioritization partial orders involved. One can view this as a kind of associativity property. Formally, using the notation from the above paragraph:

$$R1{*}(RT{*}HTT) \;\equiv\; (R1{\circ}RT){*}HTT$$

(On the right-hand-side, we treat HTT as one big tuple.) Secondly, the result of composing prioritization is well-defined. Formally: Given well-foundedness of the prioritization partial orders ($R1$ and each of the RTi's), the result of composing prioritization is a prioritized pre-order with a well-founded prioritization partial order. By Theorem 2.34, that result is, therefore, a pre-order. This well-definition of composing prioritization is maintained for 1 up to any finite number of composition steps.

Proof Overview: Uses Theorem 2.52. See Appendix. □

Proof of Theorem 2.56 :
Associativity: We want to show that:

$$R1{*}(RT{*}HTT) \;\equiv\; (R1{\circ}RT){*}HTT \tag{G}$$

Let PT stand for $RT{*}HTT$. Let R stand for $R1{\circ}RT$. Let NTk stand for the index set of PTk, the k^{th} member of the tuple PT. Let $N1$ stand for the index set of $R1$. Let N stand for the index tuple of HTT, when HTT is viewed as one big tuple (rather than as a tuple of tuples). Let $\preceq_a$ stand for $\preceq_{HTTa}$, the a^{th} member of the tuple HTT when HTT is viewed as one big tuple. Likewise, let $\preceq_b$ stand for $\preceq_{HTTb}$, the b^{th} member of the tuple HTT when HTT is viewed as one big tuple.

Elaborating the definition of $\preceq_{PTk}$:

$$\preceq_{PTk} \;\stackrel{\text{def}}{\equiv}\; \forall a \in NTk.\, \approx_{RTk_{D(a)}} \;\supset\; \preceq_a \tag{0}$$

Elaborating its definition, using PT, the left-hand-side of (G) $\;\stackrel{\text{def}}{\Longleftrightarrow}$

$$\forall i \in N1.\, [\forall j \in N1.\, R1(j,i) \supset \approx_{PTj}] \;\supset\; \preceq_{PTi} \tag{1}$$

Consider

$$\approx_{PTj} \tag{1b}$$

Using (0): (1b) $\;\Longleftrightarrow$

$$\forall a \in NTj.\, \approx_{RTj_{D(a)}} \;\supset\; \approx_a \tag{2}$$

By Lemma B.10, (2) $\;\Longleftrightarrow$

$$\forall a \in NTj.\, \approx_a \tag{3}$$

Consider

$$\preceq_{PTi} \tag{1c}$$

Using (0) and definitionally elaborating in further detail: (1c) $\overset{\text{def}}{\Longleftrightarrow}$

$$\forall b \in NTi.\ [\forall a \in NTi.\ RTi(a,b) \supset \approx_a] \ \supset \ \preceq_b \tag{4}$$

By substituting (3) for (1b) and (4) for 1c) and re-writing a bit: (1) $\Longleftrightarrow$

$$\begin{aligned} &\forall i \in N1.\ \forall b \in NTi. \\ &\qquad \{[\forall j \in N1.\ R1(j,i) \supset (\forall a \in NTj.\ \approx_a)] \\ &\qquad\quad \wedge\ [\forall a \in NTi.\ RTi(a,b) \supset \approx_a]\} \\ &\qquad\qquad \supset\ \preceq_b \end{aligned} \tag{5}$$

$\Longleftrightarrow$

$$\begin{aligned} &\forall b \in N. \\ &\quad \{\forall a \in N.\ \forall i,j \in N1. \\ &\qquad \{[(b \in NTi) \wedge (a \in NTj) \wedge R1(j,i)] \\ &\qquad\quad \vee\ [(b \in NTi) \wedge (a \in NTi) \wedge RTi(a,b)]\} \\ &\qquad\qquad \supset\ \approx_a\} \\ &\qquad \supset\ \preceq_b \end{aligned} \tag{6}$$

Using Definition 2.51 of composing partial orders implies that (6) $\Longleftrightarrow$

$$\forall b \in N.\ \{\forall a \in N.\ R(a,b)\ \supset\ \approx_a\}\ \supset\ \preceq_b \tag{7}$$

(7) is just the definition of the right-hand-side of (G). **QED Associativity**

Well-definition: We want to show that $G \overset{\text{def}}{=} (R1{*}(RT{*}HTT))$ is well-defined as a prioritized pre-order. By the associativity part of the Theorem (shown above), G is equivalent to $(R1{\circ}RT){*}HTT$. That $R1{\circ}RT$ is well-founded follows from the closure of the well-founded class of partial orders under composition: Theorem 2.52. Thus G is well-defined as a prioritized pre-order. That G is well-defined as a pre-order follows from the well-definition of prioritization: Theorem 2.34. **QED Well-definition**

QED Theorem 2.56

B.9 For section 2.8: Meetings Example

Proof of Example 2.57 :

Our arguments are directly in terms of models. By models below, we mean models that satisfy the base of the circumscription. Such an argument is concise because the example is relatively simple in terms of the conflicting interactions between the defaults. For more complicated examples, one can use our results in the following chapters, especially chapters 5 and 6, plus those in [Lifschitz, 1985].

I): We want to show that:

$PPC(B1; ab; R; Z) \models meet(Ed, Today, 4pm)$

It is straightforward to check that there is a non-empty class of models in which all of the abnormalities are identically false:

$ab = False$

Here, $False$ stands for the tuple, similar to ab, of identically false formulas.

The models in this class are exactly the set of most preferred models: no other model can "do better" than them with respect to minimizing $(ab; R)$. Thus the circumscription entails:

$ab = False$

Another way of viewing this is that all of the defaults "go through" (see discussion of this in sub-section 5.13.3).

In particular, $\neg ab3(Ed, Today, 4pm)$ is entailed,
and therefore $meet(Ed, Today, 4pm)$ is entailed.

QED I)

II): We want to show that:

$PPC(B2; ab; R; Z) \models \neg meet(Ed, Today, 4pm)$

It is straightforward to check that there is a non-empty class $\mathcal{P}$ of models in which all of the abnormalities are identically false, except that:

$$\forall p.\ ab3(p, Today, 4pm) \tag{1}$$

It is also straightforward to check that $B2$ entails:

$$\forall p.\ \neg ab3(p, Today, 4pm) \supset ab2(p, Today, 4pm) \tag{2}$$

Consider any model M in the class $\mathcal{P}$. No other model M', that is not in $\mathcal{P}$, can improve on M; here by "improve" we mean: be strictly preferred with respect to minimizing $(ab; R)$. Why not? The only opportunity for M' to improve on M is to make $ab3$ be false for some tuple $\langle e, Today, 4pm \rangle$, where e is a member of the domain for M'. But by (2), this implies that $ab2$ is true at that same tuple. Since minimizing $ab2$ has higher-priority than minimizing $ab3$, M' is strictly less preferred than M. (Actually, there is a bit of an additional subtlety: M' might have a different interpretation of the function symbols than M, in which case since we are assuming functions to be fixed, M' and M are incomparable under the global policy pre-order. But then there exists another member M'' of $\mathcal{P}$ that does agree with M' on the functions, and that satisfies exactly the same ab formulas as does M. M'' is thus strictly preferred to M'.) Thus not only is M' not strictly preferred to M, but M (actually, M'') is strictly preferred to M'. Thus all of the models outside of $\mathcal{P}$ are strictly less preferred than (some one of) the models inside of $\mathcal{P}$.

Also, no model within $\mathcal{P}$ is strictly preferred to any other within P. Thus $\mathcal{P}$ is exactly the class of most preferred models. Therefore, the circumscription entails that all of the abnormalities are identically false, except for (1).

In particular, $\neg ab3(Ed, Today, 4pm)$ is entailed,
and therefore $\neg meet(Ed, Today, 4pm)$ is entailed.

QED II)

III): We want to show that:

$$PPC(B3; ab; R; Z) \not\models meet(Ed, Today, 4pm) \tag{G1}$$

$$PPC(B3; ab; R; Z) \not\models \neg meet(Ed, Today, 4pm) \tag{G2}$$

It is straightforward to check the following three facts.
1) There is a non-empty class $\mathcal{P}_1$ of models in which all of the abnormalities are identically false, except that

$$ab3(Ed, Today, 4pm) \tag{1}$$

2) There is a non-empty class $\mathcal{P}_2$ of models in which all of the abnormalities are identically false, except that

$$ab4(Ed, Today) \tag{2}$$

3) $B3$ entails

$$ab3(Ed, Today, 4pm) \vee ab4(Ed, Today) \tag{3}$$

$\mathcal{P}_1$ and $\mathcal{P}_2$ are thus disjoint.

Fact 3) implies that no model outside of $\mathcal{P}_1 \cup \mathcal{P}_2$ can be strictly preferred to any model inside of $\mathcal{P}_1 \cup \mathcal{P}_2$. Indeed, by a similar argument to that we gave in **II)** above, each of these outside models is strictly less preferred than some model inside $\mathcal{P}_1 \cup \mathcal{P}_2$.

However, none of the models in $\mathcal{P}_1$ are strictly (or even non-strictly) preferred to any of the models in $\mathcal{P}_2$, nor vice versa. Doing better on $ab3(Ed, Today, 4pm)$ can only come at the price of doing worse on $ab4(Ed, Today)$, and vice versa. Yet, there is no strict priority between minimizing $ab3$ and minimizing $ab4$. Thus $\mathcal{P}_1 \cup \mathcal{P}_2$ is exactly the class of most preferred models.

Each of the models in $\mathcal{P}_1$ satisfies $\neg ab4(Ed, Today)$,
and therefore $\neg meet(Ed, Today, 4pm)$. This implies (G1).
Likewise, Each of the models in $\mathcal{P}_2$ satisfies $\neg ab3(Ed, Today, 4pm)$,
and therefore $meet(Ed, Today, 4pm)$. This implies (G2).

QED III)

QED Example 2.57

B.10 Basics of Positivity / Negativity

Guide to Reader: Pre-requisite: sections 2.2 and B.1.

Positivity turns out to be a key condition in analyzing monotonicity of base and default and prioritization updates.

Terminology: Positive / Negative Syntactic Appearance By positive / negative appearance, we mean the standard logical notion. We say that the predicate symbols Q appear *positively* (respectively, *negatively*) in the formula $E[Q, Y]$ when E is equivalent to a formula in which the only logical connectives are $\neg$, $\wedge$, and $\vee$, and every appearance of each member of Q appears in the scope of an even (respectively, odd) number of negations. We also say that "E is positive in Q" (respectively "negative in Q"). We also call this notion *syntactic* positivity / negativity; later (in section E.1), we will define another kind of positivity.

Fact B.11 (Axiomatic Characterization of Positive / Negative)
Let $E[Z, x]$ be an open formula in a tuple of predicates Z, and a tuple of individual object variables x. Let a tuple of predicates Q be a subset of Z. Define $Q \stackrel{\text{def}}{=} Z - Q$, so that $Z = \langle Q, Y \rangle$. The usual notion of positivity for E with respect to Q is that all syntactic appearances of Q in E are positive. (See above.) This is equivalent to the following axiomatic characterization:

$$\models \;\; Q \leq Q' \;\; \supset \;\; E[Q, Y] \leq E[Q', Y]$$

which can be elaborated as:

$$\models \;\; \forall Q, Q'.\, Q \leq Q' \;\; \supset \;\; E[Q, Y] \leq E[Q', Y]$$

which can be elaborated and re-written as:

$$\forall Z, Z'.\, Q \leq Q' \wedge Y = Y' \;\; \supset \;\; E[Z] \leq E[Z']$$

(Here, $\models$ is with respect to second-order logic.)

Negativity is defined in the same way as positivity, except that, above, the rightmost $\leq$'s are replaced by $\geq$.

Generalization: In section E.1, we will generalize the concept of positivity / negativity to be with respect to pre-orders.

B.11 For sub-section 2.9.1: Monotonicity and Discrimination

Theorem 2.58 (Prioritization is Monotonic)
Increasing the prioritization of the minimized pre-order is monotonic for any circumscription.

$$[\forall xy.\, R1(x, y) \supset R2(x, y)] \quad \Rightarrow$$

$$[C(B; (H; R2); Z) \;\models\; C(B; (H; R1); Z)]$$

where $R2$ is defined over the same indices (e.g., N) as $R1$.

By "increasing" the prioritization, we mean adding more pairs (paths) to the prioritization partial order (dag).

In the Circumscriptive Language of Defaults, defined in section 3.2, prioritization information is specified via axioms, and accumulates monotone-increasingly. Thus the above implies that:
In CLD, updating with prioritization axioms is globally monotonic.
(Moreover, in terms of Definition 4.17, this monotonicity is "forever".)

Proof Overview: Uses a positivity / negativity argument. See Appendix. □

Proof of Theorem 2.58 : Our argument is in terms of positivity and negativity.

In Definition 2.33, suppose we consider R and N as first-order predicates in the same higher-order language as Z and Z', instead of as meta-linguistic relative to Z. (For an elaborated version of this, see [Grosof, 1992a].) Then we can write Definition 2.33 equivalently as its "quantified" version:

$$Z \preceq_{(H;R)} Z' \;\stackrel{\text{def}}{\equiv}\; \forall i.\, [\forall j.\, N(j) \wedge R(j,i) \;\supset\; Z \approx_{Hj} Z'] \;\;\supset\;\; [N(i) \;\supset\; Z \preceq_{Hi} Z']$$

Examining this, we see that any prioritized pre-order $(H^N; R)$ is syntactically positive in the prioritization predicate R.
(We **presume** that H contains no appearances of R.)
We can say this more abstractly: that

$$\preceq_{R^N}$$

is positive in R. Next, we notice that Lemma B.10 $\Longrightarrow$

$$\prec_{R^N} \;\equiv\; \preceq_{R^N} \wedge \neg(\forall i \in N.\, \approx_i\,) \tag{1}$$

Since the rightmost expression in (1) is R-free, $\prec_{R^N}$ is thus positive in R. In other words, $Z \prec_{(H;R)} Z'$ is positive in R. The circumscription formula

$$B(Z) \;\wedge\; \neg\exists Z'.\, B(Z') \wedge Z' \prec_{(H;R)} Z \wedge Z' \preceq_G Z$$

is, therefore, negative in R. This negativity plus Fact B.11 implies the Theorem. **QED Theorem 2.58**

We find it convenient, for later proofs, to re-express the essence of Theorem 2.58 and its proof in a different notation.

Lemma B.12
Let S and R be prioritization partial orders, where R is a refinement of S:

$$S \leq R \tag{0}$$

Then

$$\models\;\; \preceq_S \;\supset\; \preceq_R$$

$$\models \ \prec_S \ \supset \ \prec_R$$

$$\models \ \neg\prec_R \ \supset \ \neg\prec_S$$

In particular, when $S = \emptyset$:

$$\models \ \preceq_\emptyset \ \supset \ \preceq_R$$

$$\models \ \prec_\emptyset \ \supset \ \prec_R$$

In more detail: Let H be a tuple of pre-orders, with index set N, defined over Z, and let S and R be defined over N. Then

$$\models \ Z \preceq_{(H;S)} Z' \ \supset \ Z \preceq_{(H;R)} Z'$$

$$\models \ Z \prec_{(H;S)} Z' \ \supset \ Z \prec_{(H;R)} Z'$$

Proof : Immediate from Theorem 2.58 and its proof. **QED**

Theorem 2.59 (Discrimination)
Let D be a finite tuple of default formulas, indexed by N, in symbols Z, that are satisfiable and logically independent of each other. Let $R1$ and $R2$ be two distinct (different) prioritization partial orders defined over N. Then there exists a satisfiable base sentence $B[Z]$ that results in non-equivalent default theories for the two prioritizations; i.e., such that

$$PDC(B; D; R1; Z) \ \not\Leftrightarrow \ PDC(B; D; R2; Z)$$

Proof Overview: Constructive. See Appendix. $\square$

Proof of Theorem 2.59 :

For simplicity's sake, we assume that the default formulas are all sentences, i.e., closed formulas. The proof easily generalizes to the case when they may be open; we discuss how at the end of the proof.

Since $R1$ and $R2$ are different, they must differ about at least one pair (i, j) (where $i, j \in N$). Consider that pair. There are two possible kinds of cases. **I)** $R1$ and $R2$ are diametrically opposed about whether i is higher priority than j; one (say, $R1$) says it is, and the other says i has lower priority than j. **II)** One of the two prioritizations (say, $R1$) is definite about whether i is higher priority than j (say, it is indeed higher priority), while in the other i and j are incomparable (i.e., neither has higher priority).

Since the default formulas $Dk[Z]$ are logically independent, we can choose B to be any Boolean combination of them. Let B include, first of all, the negation of all the defaults except for the i^{th} and the j^{th}. That is, let B include the conjunction

$$\forall k \neq i, j \in N. \ \neg Dk[Z]$$

Let B include, furthermore, the sentence

$$\neg Di[Z] \vee \neg Dj[Z]$$

The intuition is that B guarantees that the i^{th} default and the j^{th} default conflict.

I) For definiteness, assume $R1(i,j) \wedge R2(j,i)$. Then $R1$ makes the i^{th} default "win" and the j^{th} default "lose", while $R2$ does the reverse.

$$PDC(B;D;R1;Z) \equiv \{B[Z] \wedge Di[Z] \wedge \neg Dj[Z]\}$$

$$PDC(B;D;R2;Z) \equiv \{B[Z] \wedge \neg Di[Z] \wedge Dj[Z]\}$$

These are not equivalent; they are mutually inconsistent.

II) For definiteness, assume $R1(i,j) \wedge \neg R2(i,j) \wedge \neg R2(j,i)$. Then $R1$ makes the i^{th} default "win" and the j^{th} default "lose", while $R2$ is "cautious" and lets neither "win".

$$PDC(B;D;R1;Z) \equiv \{B[Z] \wedge Di[Z] \wedge \neg Dj[Z]\}$$

$$PDC(B;D;R2;Z) \equiv \{B[Z] \wedge (Di[Z] \equiv \neg Dj[Z])\}$$

These are not equivalent; the first is strictly stronger than (i.e., implies, and is not implied by) the second.

To generalize to the case where the default formulas are open instead of closed: use the universal closures of the default formulas when constructing B in the manner above. That is, $\forall xh.\ Dh[Z, xh]$ plays the role of $Dh[Z]$ above, for $h = i, j, k$.

Terminology: By *universal closure*, we mean adding a universal quantifier for each free (individual / object) variable in Ei.

E.g., if Ei is $governor(x,y,t) \supset governor(x,y,t+1)$, then its universal closure is $\forall x,y,t.\ governor(x,y,t) \supset governor(x,y,t+1)$. If Ei is closed, then it is its own universal closure.

QED Theorem 2.59

B.12 For sub-section 2.9.2: Embedding Inheritance

Proof of Example 2.60 :

We treat a simplified version of the example, that retains its essence. We formulate the example in terms of propositional language, in effect restricting the domain to just Tom and $Peter$. The example can be formulated in full pointwise detail using the representational generalization given in [Grosof, 1992a], then analyzed using the results in [Grosof, 1992c].

We define and show the propositional PPC version of the example as follows.

By conservative extension for abnormalities, Theorem 3.8[5], it suffices to show the propositional PDC version has the desired entailments and non-entailments.

[5] In this proof Appendix, we generally avoid using results that appear, sequentially, later in the Appendix. This is an exception. However, there is no circularity. Example 2.60 is not used to prove a result anywhere else in this dissertation.

We define this propositional PDC version as follows. We make each atom into a primitive propositional letter:

sp stands for $student(Peter)$.

st stands for $student(Tom)$.

ap stands for $adult(Peter)$.

at stands for $adult(Tom)$.

wp stands for $work(Peter)$.

wt stands for $work(Tom)$.

The tuple Z of all predicate symbols in the base language, is thus:

$$\langle sp, st, ap, at, wp, wt\rangle$$

(There are no function symbols, since the base language is propositional.)

The base axioms are:

$$sp \supset ap$$
$$st \supset at$$
$$ap$$
$$st$$
$$wp \equiv wt$$

The default formulas D^N are:

$$D1p \stackrel{\text{def}}{\equiv} ap \supset wp$$
$$D1t \stackrel{\text{def}}{\equiv} at \supset wt$$
$$D2p \stackrel{\text{def}}{\equiv} sp \supset \neg wp$$
$$D2t \stackrel{\text{def}}{\equiv} st \supset \neg wt$$

The index tuple N is thus defined as: $\langle 1p, 1t, 2p, 2t\rangle$.

We recommend as appropriate for this example a non-layered prioritization partial order R. The relation R contains exactly the pairs:

$$(2p, 1p)$$
$$(2t, 1t)$$

The layered prioritization that we regard as inappropriate for this example is $R2$, where the relation $R2$ contains exactly the pairs:

$$(2p, 1p)$$
$$(2p, 1t)$$
$$(2t, 1p)$$
$$(2t, 1t)$$

We give an argument directly in terms of models. By models below, we mean models that satisfy the base B.

We want to show that:

$$PDC(B; D; R; Z) \not\models wp \tag{G1}$$

$$PDC(B; D; R; Z) \not\models \neg wp \tag{G2}$$

$$PDC(B;D;R;Z) \not\models wt \tag{G3}$$

$$PDC(B;D;R;Z) \not\models \neg wt \tag{G4}$$

and that:

$$PDC(B;D;R2;Z) \models \neg wp \wedge \neg wt \tag{G5}$$

The base B leaves three propositional letters indeterminate: sp, wp, and st; however, it constrains wp and wt to be equivalent. There are thus exactly four models, which we characterize as:

Model	sp	wp	wt	satisfied defaults
$M1:$	$+$	$+$	$+$	$1p, 1t$
$M2:$	$+$	$-$	$-$	$2p, 2t$
$M3:$	$-$	$+$	$+$	$1p, 1t, 2p$
$M4:$	$-$	$-$	$-$	$2p, 2t$

Here, each row corresponds to a model. $+$ in the column corresponding to a letter means that that letter is satisfied in the model; $-$ means that it is not. The last column indicates which default formulas are satisfied in that model.

It is straightforward, using the definition of prioritized default pre-orders, to show the following preferences among these models.

Preferences with the layered prioritization: i.e., with respect to maximizing $(D; R2)$:
$M2$ is strictly preferred to $M1$: though $M1$ does better on defaults $1p, 1t$, these are lower priority than the defaults $2p, 2t$.
$M4$ is equivalent to $M2$: it satisfies exactly the same set of defaults. ($M4$ is thus, likewise, strictly preferred to $M1$.)

$M2$ is also strictly preferred to $M3$: though $M3$ does better on $1p, 1t$, $M2$ does better on $2t$, which has higher priority than both $1p$ and $1t$. (Likewise, $M4$ is strictly preferred to $M3$.)

Thus the set of maximally preferred models is: $\{M2, M4\}$. Therefore, the circumscriptive augmentation is equivalent to:

$\neg wp \wedge \neg wt$

This implies (G5).

Preferences with the non-layered prioritization: i.e., with respect to maximizing $(D; R)$:
$M2$ is, again, strictly preferred to $M1$.
$M4$ is still equivalent to $M2$.

However, $M3$ is incomparable with $M2$, and also with $M4$: there is no strict (or even non-strict) preference either way. This is because $2t$ does not have higher priority than $1p$. $M3$ is also strictly preferred to $M1$: it does better on $2p$.

Thus $M3$ is a maximally preferred model, along with $M2$ and $M4$. Therefore, the circumscriptive augmentation is equivalent to:

$(sp \wedge \neg wp \wedge \neg wt) \vee (\neg sp \wedge wp \wedge wt) \vee (\neg sp \wedge \neg wp \wedge \neg wt)$

This implies (G1) through (G4).

Discussion: The non-layered prioritization relation includes fewer (strict-priority) pairs than the layered prioritization relation. This results in a strictly weaker circumscription, yielding the desired non-entailments.

QED Example 2.60

B.13 For sub-section 2.11.1: Alternative Definition of Prioritization

Theorem 2.69 (Equivalence of Definition #2 of Prioritization)
Let R be well-founded. Then

$$Z \preceq_{(H;R)} Z' \;\;\equiv\;\; \forall i \in N.\, [\forall j \in N.\, R(j,i) \supset \neg(Z \prec_{Hj} Z')] \;\supset\; (Z \preceq_{Hi} Z')$$

One can read the right hand side as: "each voter votes non-strictly yes if her vote is not obviated by a strict yes vote by some individual higher priority voter".

Proof Overview: Complicated, inductive on R. See Appendix. □

Proof of Theorem 2.69 :

This proof is complicated, and for the sake of clarity and brevity, we introduce a lot of local notation.

Local notation and terminology:

Let us call Definition 2.33 of prioritization: Definition #1. We will make this explicit by writing $\preceq_{H;R}$ as:

$\preceq^{\mathbf{1}} (H;R)$

Let N be the index tuple of H, and thus the domain of R. Our discussion below does not depend on the details of H^N. And we will be considering only one precedence p.o.: R. Rather, we will emphasize the role of N in the notation. Henceforth, we will write the above as:

$\preceq^{\mathbf{1}} (N)$

Let us call the definition in the right-hand-side of the Theorem: Definition #2. We will write this binary relation (we have yet to show it to be a pre-order) as:

$\preceq^{\mathbf{2}} (H;R)$

or, henceforth, as:

$\preceq^{\mathbf{2}} (N)$

Let

$L_j^1 \overset{\text{def}}{\equiv} \neg\approx_j$

$L_j^2 \overset{\text{def}}{\equiv} \prec_j$

We reserve the symbol h as an index in the proof, that ranges over the domain $\{1, 2\}$. That is, h indexes the definition #.

Henceforth: all appearances of indices i, j, k, l are members of N; and all index sets (e.g., U, V, Si, $D(i)$) are subsets of N. We will often leave this implicit.

We employ our usual abbreviated notation for prioritization partitions, and our usual notation for projections and selected segments of prioritized pre-orders: recall section B.5. In addition, we extend the concepts of projections selected segments to Definition #2, and we abbreviate the projection and selected segment notation to omit mention of R.

For any index set U, we thus define projection onto U as:

$$\preceq^{\mathbf{h}}(U) \ \stackrel{\text{def}}{\equiv}\ \forall i \in U.\ \preceq_i \vee (\exists j \in U.\ R(j,i) \wedge L_j^h)$$

For any index sets U and V such that $V \subseteq U$, we thus define the segment of the projection U selected by V as:

$$\preceq^{\mathbf{h}}(U:V) \ \stackrel{\text{def}}{\equiv}\ \forall i \in V.\ \preceq_i \vee (\exists j \in U.\ R(j,i) \wedge L_j^h)$$

In the usual way, we define "tie" and strict versions of the above, e.g.:
$\approx^{\mathbf{h}}(U) \ \stackrel{\text{def}}{\equiv}\ (\preceq^{\mathbf{h}}(U) \wedge \succeq^{\mathbf{h}}(U))$ and $\approx^{\mathbf{h}}(U) \ \stackrel{\text{def}}{\equiv}\ (\preceq^{\mathbf{h}}(U) \wedge \neg\succeq^{\mathbf{h}}(U))$.

And, as usual, we write $R_D(i)$ as: $D(i)$.

Let

$$Si \ \stackrel{\text{def}}{\equiv}\ D(i) \cup \{i\}$$

We say that Si is the *"up-closure"* of i with respect to R and N.

So much for notation and terminology. Next, we give two local lemmas. The second involves the main inductive part of the overall proof of the Theorem.

Local Lemma 1: Properties of Si:

$$Si \text{ is up-closed (Definition B.7).} \tag{LL1.1}$$

$$Sk \subseteq N \tag{LL1.2}$$

$$R(k,i) \ \supset \ Sk \subseteq Si \tag{LL1.3}$$

$$N \ = \ \textstyle\bigcup_{k \in N} Sk \tag{LL1.4}$$

$$D(i) \ = \ \textstyle\bigcup_{k \in D(i)} Sk \tag{LL1.5}$$

$$\forall l \in Sk.\ D(l) \subseteq Sk \tag{LL1.6}$$

Proof of Local Lemma 1 : Each of these properties follows easily from the definition of Si, the transitivity of R, and the properties of up-closed-ness, including: Lemma E.9 parts (1) and (2), and Lemma E.10 part (1). **QED Local Lemma 1**

Recall also Lemmas E.8 and E.23 as **background**.

Local Lemma 2:

$$\forall i \in N. \quad \preceq^{\mathbf{2}}(Si) \ \equiv \ \preceq^{\mathbf{1}}(Si) \tag{LL2}$$

Proof of Local Lemma 2 : By well-founded induction: on i, using the precedence partial order R, in the direction of descending precedence.

Assume the inductive hypothesis is true for all $k \in N \ni\!\!-R(k,i)$, i.e., for all $k \in D(i)$:

$$\forall k \in D(i). \quad \preceq^{\mathbf{2}} (Sk) \;\equiv\; \preceq^{\mathbf{1}} (Sk) \tag{1}$$

Then it suffices to show the inductive hypothesis is true for i:

$$\preceq^{\mathbf{2}} (Si) \;\equiv\; \preceq^{\mathbf{1}} (Si) \tag{G1}$$

Local Lemma 1 (LL1.3) and (LL1.6) $\Longrightarrow$

$$\forall k \in D(i). \;\; \forall h. \;\; \preceq^{\mathbf{h}} (Sk) \equiv \preceq^{\mathbf{h}} (Si : Sk) \tag{2}$$

(1) and (2) $\Longrightarrow$

$$\forall k \in D(i). \quad \preceq^{\mathbf{2}} (Si : Sk) \equiv \preceq^{\mathbf{1}} (Si : Sk) \tag{3}$$

(3) $\Longrightarrow$

$$(\forall k \in D(i). \preceq^{\mathbf{2}} (Si : Sk)) \;\equiv\; (\forall k \in D(i). \preceq^{\mathbf{1}} (Si : Sk)) \tag{4}$$

For any formula $E[x,i]$, Local Lemma 1 (LL1.5) $\Longrightarrow$

$$(\forall k \in D(i). \, E[k,i]) \;\equiv\; (\forall k \in D(i). \, \forall l \in Sk. \, E[l,i]) \tag{5}$$

In particular, choose $E[x,i]$ to be:

$\preceq_x \;\vee\; \exists j \in D(x) \cap Si. \, L_j^h$

Thus, (5) $\Longrightarrow$

$$\forall h. \quad \preceq^{\mathbf{h}} (Si : D(i)) \;\equiv\; (\forall k \in D(i). \preceq^{\mathbf{h}} (Si : Sk)) \tag{6}$$

(4) and (6) $\Longrightarrow$

$$\preceq^{\mathbf{2}} (Si : D(i)) \;\equiv\; \preceq^{\mathbf{1}} (Si : D(i)) \tag{7}$$

The definition of Si $\Longrightarrow$

$$\forall h. \quad \preceq^{\mathbf{h}} (Si) \;\equiv\; (\preceq^{\mathbf{h}} (Si : D(i)) \;\wedge\; \preceq^{\mathbf{h}} (Si : \{i\})) \tag{8}$$

Next, we use the tautology that, for any propositions $a1, a2, b1, b2$:

$$\{(a1 \equiv b1) \;\wedge\; [(a1 \wedge b1) \;\supset\; (a2 \equiv b2)]\} \;\supset\; [(a1 \wedge a2) \;\equiv\; (b1 \wedge b2)] \tag{Taut}$$

Assume

$$\preceq^{\mathbf{2}} (Si : D(i)) \tag{9}$$

and also assume

$$\preceq^{\mathbf{1}} (Si : D(i)) \tag{10}$$

Then (7), (8), and (Taut) imply that it suffices to show

$$\preceq^{\mathbf{2}} (Si : \{i\}) \;\equiv\; \preceq^{\mathbf{1}} (Si : \{i\}) \tag{G2}$$

Definitionally elaborating, the left-hand-side of (G2) $\stackrel{\text{def}}{\Longleftrightarrow}$

$$\preceq_i \;\vee\; (\exists j \in Si.\, R(j,i) \wedge \prec_j) \tag{G2a}$$

Likewise, the right-hand-side of (G2) $\stackrel{\text{def}}{\Longleftrightarrow}$

$$\preceq_i \;\vee\; (\exists j \in Si.\, R(j,i) \wedge \neg\approx_j) \tag{G2b}$$

Therefore, using the irreflexivity of R and the definition of Si: it suffices to show

$$(\exists j \in D(i).\, \prec_j) \;\equiv\; (\exists j \in D(i).\, \neg\approx_j) \tag{G3}$$

In (G3), the left-to-right implication follows immediately from the fact that:

$\models\; \prec_j \;\supset\; \neg\approx_j$

For the right-to-left implication in (G3):
Assume the right-hand-side of (G3):

$$\exists j \in D(i).\, \neg\approx_j \tag{11}$$

Then it suffices to show the left-hand-side of (G3):

$$\exists j \in D(i).\, \prec_j \tag{G4}$$

(10) and Local Lemma 1 (LL1.6) $\Longrightarrow$

$$\preceq^{\mathbf{1}} (D(i)) \tag{12}$$

(11), Lemma E.8, and the contraposition of Lemma B.10 $\Longrightarrow$

$$\neg\approx^{\mathbf{1}} (D(i)) \tag{13}$$

Here, we have let $D(i)$ play the role of N in that Lemma B.10, and we have let the restriction of R to $D(i)$, i.e., $R^{D(i)}$, play the role of R in that Lemma. Lemma E.8 tells us that is OK.
(11), (12), and (13) $\Longrightarrow$

$$\approx^{\mathbf{1}} (D(i)) \tag{14}$$

Lemma E.8 and Lemma E.13[6] $\Longrightarrow$

$$\approx^{\mathbf{1}} (D(i)) \;\equiv\; (\preceq^{\mathbf{1}} (D(i)) \;\wedge\; \exists j \in D(i).\, \prec_j) \tag{15}$$

[6]Lemma E.13 is at the very end of section E.3. In this proof Appendix, we generally avoid using results that appear, sequentially, later in the Appendix. This is an exception. However, there is no circularity of dependence among our results and proofs: no need to worry! Theorem 2.69 is used only once to prove a result: in section 2.11.1, to show Theorem 2.68. Neither Theorem 2.69 nor Theorem 2.68 are used anywhere else in this dissertation to prove a result.

Here, we have let $D(i)$ play the role of N in Lemma E.13, and we have let the restriction of R to $D(i)$, i.e., $R^{D(i)}$, play the role of R in that Lemma. As in the step leading to (13) above, Lemma E.8 tells us that is OK.

(14) and (15) $\implies$ (G4).

QED Local Lemma 2

The proof proper of the Theorem:

We want to prove that Definition #2 is equivalent to Definition #1; i.e., that:

$$\preceq^{\mathbf{1}} (N) \;\equiv\; \preceq^{\mathbf{2}} (N) \tag{G}$$

For any formula $E[l]$, Local Lemma 1 (LL1.4) $\implies$

$$(\forall l \in N.\, E[l]) \;\equiv\; (\forall i \in N.\, \forall l \in Si.\, E[l]) \tag{1}$$

In particular, choose $E[l]$ to be:

$\preceq_l \;\vee\; (\exists j \in N.\, R(j, l) \wedge L_j^h)$

Thus, (1) $\implies$

$$\forall h. \quad \preceq^{\mathbf{h}} (N) \;\equiv\; (\forall i \in N.\, \preceq^{\mathbf{h}} (N : Si)) \tag{2}$$

Local Lemma 1 (LL1.2) and (LL1.6) $\implies$

$$\forall h. \quad \preceq^{\mathbf{h}} (N : Si) \;\equiv\; \preceq^{\mathbf{h}} (Si) \tag{3}$$

(The crux here is: because Si is up-closed.)

Local Lemma 2 (LL2) $\implies$

$$(\forall i \in N.\, \preceq^{\mathbf{2}} (Si)) \;\equiv\; (\forall i \in N.\, \preceq^{\mathbf{1}} (Si)) \tag{4}$$

(2), (3), and (4) $\implies$ (G).

QED Theorem 2.69

Appendix C

Proofs for Chapter 3

C.1 For section 3.2: Conservative Extension for Abnormalities

Lemma C.1
Let Y be a tuple of predicate symbols, with index set N. Let L and U be tuples of formulas that are Y-free. Furthermore, let L and U each be similar to Y. Let R be a prioritization p.o. that is defined over N. (Note that R could, for example, be $\emptyset$.)
Then

$$[\exists Y.\, L{=}Y{\prec_R}U] \;\;\equiv\;\; [L{\prec_R}U]$$

Proof : From left to right: use reflexivity and transitivity of prioritized predicate pre-orders. From right to left: substitute L for Y, and use reflexivity of prioritized predicate pre-orders. **QED**

Lemma C.2
Let Y be a tuple of predicate symbols, with index set N. Let L and U be tuples of formulas that are Y-free. Furthermore, let L and U each be similar to Y. Let S and R be prioritization p.o.'s, defined over N, where R is a refinement of S:

$$S \le R \tag{0}$$

(Note that S for example, be $\emptyset$, and that R could, for example, be the same as S.)
Then

$$[\exists Y.\, L{\preceq_S}Y{\prec_R}U] \;\;\equiv\;\; [L{\prec_R}U]$$

Proof : Same as for Lemma C.1, but also use Lemma B.12 for the left-to-right direction. **QED**

Theorem 3.8 (Conservative Extension, Abnormality Theories)
An abnormality theory in either the adaptive or the definitional forms given in Definition 3.5 is equivalent to:

$$(\neg ab{=}D[Y]) \;\wedge\; PDC(B; D; R; Y)$$

i.e., is equivalent to:

$$(ab{=}E[Y]) \ \wedge \ PFC(B;E;R;Y)$$

where D and $\neg ab$ are defined as in Definition 3.5.

In other words, the abnormality-style prioritized *predicate* circumscription is equivalent to the prioritized *default* circumscription that corresponds to maximizing $D[Y]$ (according to the prioritization R), conservatively extended by the explicit definitions of the ab predicates. Equivalently, it corresponds to minimizing $E[Y]$, in the same sense. (Recall the relationship, in Definition 2.43 of prioritized default circumscription to prioritized formula circumscription.)

Thus the ab-free conclusions of the abnormality theory (in either form) are exactly the same as those of the prioritized default circumscription.

$$\begin{aligned} PDC(B;D;R;Y) &\equiv \exists ab.\, PPC(B[Y] \wedge (\neg ab \leq D[Y]); ab; R; Y, ab) \\ PDC(B;D;R;Y) &\equiv \exists ab.\, PPC(B[Y] \wedge (\neg ab{=}D[Y]); ab; R; Y, ab) \end{aligned}$$

$$\begin{aligned} PFC(B;E;R;Y) &\equiv \exists ab.\, PPC(B[Y] \wedge (E[Y] \leq ab); ab; R; Y, ab) \\ PFC(B;E;R;Y) &\equiv \exists ab.\, PPC(B[Y] \wedge (E[Y]{=}ab); ab; R; Y, ab) \end{aligned}$$

Here the second-order quantifier $\exists ab$ accomplishes the projection of the prioritized predicate theory, defined over the "extended" language corresponding to the symbols $\langle Y, ab\rangle$, onto the "original" (sub-)language corresponding to the symbols $\langle Y\rangle$. Thus we can regard an abnormality theory as a prioritized default theory. (On page 101, we discuss the extension of this result to the case where some predicates may be fixed. However, that is inessential expressively, as we will show later.)

Forever Equivalence: See the end of section 4.7 for an interpretation of this result in terms of "forever equivalence", under updating, between the corresponding CLD axiom sets.

Fixtures: See Corollarỹefcorol:cons-ext-ab-fix for a generalization to prioritized default / predicate circumscriptions that include fixing.

Proof Overview: Non-trivial. Uses lemmas about removing second-order existential quantifiers and properties of prioritized pre-orders. See Appendix. □

Proof of Theorem 3.8 : We prove the easier, definitional case first, then a more general version of the adaptive case. We use the PFC version rather than the PDC version because it makes the mathematics a bit clearer.

I) Definitional case: We want to show that

$$PPC(B(Y) \wedge E(Y){=}ab; ab; R; Y, ab) \tag{1}$$

is equivalent to

$$ab{=}E(Y) \ \wedge \ PFC(B;E;R;Y) \tag{2}$$

(1) $\overset{\text{def}}{\Longleftrightarrow}$

$$B(Y) \wedge E(Y){=}ab \wedge \neg\exists Y'.\ \exists ab'.\ B(Y') \wedge E(Y'){=}ab' \wedge ab'{\prec_R}ab \tag{3}$$

Since ab is new, ab' does not appear in B(Y'), so (3) $\iff$

$$B(Y) \wedge E(Y){=}ab \wedge \neg\exists Y'.\ B(Y') \wedge \exists ab'.\ E(Y'){=}ab' \wedge ab'{\prec_R}ab \tag{4}$$

By Lemma C.1, (4) $\iff$

$$B(Y) \wedge E(Y){=}ab \wedge \neg\exists Y'.\ B(Y') \wedge E(Y'){\prec_R}ab \tag{5}$$

Using the $E(Y){=}ab$ to substitute, (5) $\iff$

$$B(Y) \wedge E(Y){=}ab \wedge \neg\exists Y'.\ B(Y') \wedge E(Y'){\prec_R}E(Y) \tag{6}$$

which is just the elaborated definition of (2). **QED I)**

II) The adaptive case: We want to show that

$$PPC(B(Y) \wedge E(Y) \leq ab; ab; R; Y, ab) \tag{1}$$

is equivalent to

$$ab{=}E(Y) \quad \wedge \quad PFC(B; E; R; Y) \tag{2}$$

We show a stronger result: Let S be any prioritization partial order that is weaker than R, i.e., such that R refines (extends) S (Definition B.8). I.e., let $S \leq R$. E.g., $S = \emptyset$. Then

$$PPC(B(Y) \wedge E(Y){\preceq_S}ab; ab; R; Y, ab) \tag{3}$$

is equivalent to (2).
(3) $\overset{\text{def}}{\iff}$

$$B(Y) \wedge E(Y){\preceq_S}ab \wedge \neg\exists Y'.\ \exists ab'.\ B(Y') \wedge E(Y'){\preceq_S}ab'{\prec_R}ab \tag{4}$$

By Lemma B.12, $S \leq R \implies$

$$\preceq_S \supset \preceq_R \tag{5}$$

Thus, by Lemma C.2, and using the new-ness of ab, (4) $\iff$

$$B(Y) \wedge E(Y){\preceq_S}ab \wedge \neg\exists Y'.\ B(Y') \wedge E(Y'){\prec_R}ab \tag{6}$$

Using the $E(Y){=}ab$ to substitute, (2) $\iff$

$$B(Y) \wedge E(Y){=}ab \wedge \neg\exists Y'.\ B(Y') \wedge E(Y'){\prec_R}ab \tag{7}$$

Comparing (7) and (6):
Clearly (7) $\implies$ (6), by reflexivity of prioritized predicate pre-orders.
To show that (6) $\implies$ (7), it suffices to show (6) $\implies$ $\neg(E(Y){\prec_S}ab)$. Substituting Y for Y' in (6), (6) $\implies$

$$B(Y) \wedge \neg(B(Y) \wedge E(Y) \prec_R ab) \tag{8}$$

$\Longrightarrow$

$$\neg(E(Y) \prec_R ab) \tag{9}$$

which, by (5) and Lemma B.12, indeed $\Longrightarrow \neg(E(Y) \prec_S ab)$. **QED II)**

The equivalence of the existential projections in the Theorem statment follows from the simpler case of Lemma C.3.

QED Theorem 3.8

C.2 For section 3.4: Quasi-Propositionality Implies Well-Behavior

Lemma 3.23 (Quasi-Propositional Implies: Open As Instances)
The sufficient condition in Theorem 2.47 can be generalized to quasi-propositionality (Definition 3.22). **Proof Overview:** See Appendix. □

Proof of Lemma 3.23 : In Definition 3.22:
Let Z be the tuple of all predicate and function symbols in the base language $\mathcal{L}$. Let NO index the open defaults. Consider an arbitrary open default $Dj[Z, xj]$. The given domain closure sentence for this open default is:

$$B[Z] \models \forall xj.\, Dj[Z, xj] \supset (\bigvee_{i=1}^{mj} xj = uji[Wj]) \tag{0.1}$$

Here each uji is a tuple, of ground terms, that has the same arity as xj has. uji mentions the function symbols $Wj \subseteq Z$: we have shown them explicitly above.

Let Ej stand for the pre-order that results from the parallel prioritization of all of the default instances of the open default $Dj[Z, xj]$:

$Z \preceq_{Ej} Z' \stackrel{\text{def}}{\equiv} \bigwedge_{i=1}^{mj} Dj[Z, uji[Wj]] \supset Dj[Z', uji[Wj']]$

Assume both

$$B[Z] \tag{1.1}$$

and

$$B[Z'] \tag{1.2}$$

Then, just as in the proof of Theorem 2.47, it suffices to show that

$$Z \preceq_{Dj} Z' \;\equiv\; Z \preceq_{Ej} Z' \tag{G1.1}$$

and also that

$$Z \succeq_{Dj} Z' \;\equiv\; Z \succeq_{Ej} Z' \tag{G1.2}$$

(0.1) and (1.1) $\Longrightarrow$

$$\forall xj.\, Dj[Z, xj] \;\supset\; (\bigvee_{i=1}^{mj} xj = uji[Wj]) \tag{2}$$

(0.1) and (1.2) $\implies$

$$\forall xj.\, Dj[Z', xj] \;\supset\; (\bigvee_{i=1}^{mj} xj = uji[Wj']) \tag{3}$$

Let $W \stackrel{\text{def}}{=} \bigcup_{j \in NO} Wj$. The given fixed-relative condition on the functions implies that W can, equivalently, be fixed in the circumscription. (See discussion in Definition 3.47.) We choose to so fix W:

$$W = W' \tag{0.2}$$

(2) $\implies$

$$(\forall xj.\, D[Z, xj] \supset Dj[Z', xj]) \;\equiv\; (\bigwedge_{i=1}^{mj} Dj[Z, uji[Wj]] \supset Dj[Z', uji[Wj]]) \tag{4}$$

Likewise, (3) $\implies$

$$(\forall xj.\, D[Z, xj] \supset Dj[Z', xj]) \;\equiv\; (\bigwedge_{i=1}^{mj} Dj[Z, uji[Wj]] \supset Dj[Z', uji[Wj]]) \tag{5}$$

(0.2) and (4) $\implies$ (G1.1).
(0.2) and (5) $\implies$ (G1.2).

QED Lemma 3.23

Theorem 3.24 (Quasi-Propositional Implies Well-Behavior)
Suppose $PDC(B; D; R; \mathit{fix}\ F; Z)$ is quasi-propositional (Definition 3.22).
Then B is **well-founded** with respect to the prioritized default pre-order $(D; R; \mathit{fix}\ F)$. Therefore, B is also **"well-founded"** with respect to $(D; R; \mathit{fix}\ F)$. Therefore, the circumscription preserves **satisfiability**.

Proof Overview: Using Lemma 3.23, we show that quasi-propositionality leads to the finiteness of the set of discriminations that the maximized pre-order can make. This finiteness implies well-foundedness under that pre-order. By Observations 3.19 and 3.17, it follows that the circumscription is also "well-founded" and satisfiable. See Appendix for details. □

Proof of Theorem 3.24 : By Observations 3.19 and 3.17, if the circumscription is well-founded, then it is also "well-founded" and satisfiable. Thus it suffices to show the circumscription is well-founded.

Let H stand for the globally maximized pre-order. We want to show that there are no infinite ascending chains of models, within the space of modules that satisfy B, where, by "ascending", we mean: strictly increasing with respect to $\sqsubseteq_H$, i.e., increasing with respect to $\sqsubset_H$.

Theorem 2.19 $\implies$

$$\sqsubset_H \;\equiv\; \sqsubset_{D;R} \wedge \cong_F$$

Therefore, it suffices to show there are no infinite ascending chains of models, within the space of models that satisfy B, where, by "ascending", we now mean: strictly increasing with respect to $\sqsubseteq_{D;R}$, i.e., increasing with respect to $\sqsubset_{D;R}$. That is, it suffices to show that $B[Z]$ is well-founded with respect to $(D; R)$.

The key observation is that it suffices to show that there are only a finite number of strict discriminations that $\sqsubseteq_{D;R}$ can make. By Fact B.3, it suffices to show the finiteness of the set of equivalence classes for $\cong_{D;R}$ over the domain of models of B.

Lemma B.10 $\Longrightarrow$

$\cong_{D;R} \;\equiv\; \bigwedge_{i \in N} \cong_{Di}$

where N stands for the index tuple of D. Therefore, it suffices to show the finiteness of the set of equivalence classes for each $\cong_{Di}$, $i \in N$, over the domain of models of B.

Case I): All defaults are closed. We begin by showing the case where all defaults are closed.

Since Di is a closed sentence, the set of equivalence classes for $\cong_{Di}$ is very simple: it has size two. In the first equivalence class, $Di[Z]$ is satisfied; in the second, it is not. **QED I)**

Case II) Quasi-propositionality: By Lemma 3.23, quasi-propositionality implies that each open default is equivalent to the parallel prioritization of the tuple of all its instances. But each instance default is closed. Thus, in the context of the circumscription, $(D;R)$ is equivalent to a prioritized default pre-order G in which each default is closed. By case **I)**, $B[Z]$ is well-founded with respect to G. Therefore, $B[Z]$ is well-founded with respect to $(D;R)$. **QED II)**

QED Theorem 3.24

C.3 For sub-section 3.5.3: Immunity of the Fixed

Lemma C.3

Let Q, P, and Y be distinct tuples of (predicate and function) symbols. Let A be a formula mentioning them. Then

$$[\exists Q.\, (Q{=}P) \;\wedge\; A(Q,P,Y)] \;\equiv\; A(P,P,Y)$$

In particular, as a special case:

$$[\exists Q.\, (Q{=}P) \;\wedge\; A(Q,Y)] \;\equiv\; A(P,Y)$$

Proof : It suffices to show the first case where A has three arguments. From right-to-left: substitute P for Q in the first argument of A. The left-to-right direction $\Longleftrightarrow$

$$\forall Q.\, [(Q{=}P) \;\wedge\; A(Q,P,Y)] \;\supset\; A(P,P,Y)$$

which follows, again, from substitution of P for Q. **QED**

Theorem 3.38 (Immunity of the Fixed, Given "Well-Foundedness")

Let $B[Z]$ be "well-founded" (see Definition 3.16) with respect to a pre-order H that fixes the symbols $W \subseteq Z$:

$$Z \preceq_H Z' \;\stackrel{\text{def}}{\equiv}\; Z \preceq_L Z' \wedge (W = W')$$

Let E[W] be any sentence purely in the fixed symbols W. Then

$$C(B;H;Z) \models E[W] \quad\Longleftrightarrow\quad B[Z] \models E[W]$$

Terminology: We call this behavior *immunity of the fixed.*
Note that, in this result, we do not make the assumption[1] that all function symbols are fixed.

Proof Overview: Not terribly complicated. Uses a lemma about removal of second-order existential quantifiers. See Appendix. □

Proof of Theorem 3.38 : Let $Y \stackrel{\text{def}}{=} Z - W$. Comparing the antecedents of the entailments: it suffices to show that

$$\forall W.\ [\exists Y.\ B[W, Y]] \equiv [\exists Y.\ C(B; H; W, Y)] \tag{1}$$

In (1), the implication from right to left is easy: it follows from the fact that $C(B)$ always entails B. Thus, by Lemma C.3, it suffices to show

$$\forall W^1, Y^1.\ B[W^1, Y^1] \ \supset\ [\exists W^2, Y^2.\ W^2 = W^1 \wedge C(B; H; W, Y)[W^2, Y^2]\,] \tag{2}$$

which follows directly from the given "well-foundedness" condition, using the definition of H.

QED Theorem 3.38

Theorem 3.39 (Immunity of the Fixed for PDC)
Any prioritized default circumscription that satisfies either of the two following well-behavior conditions also exhibits immunity of the fixed:

1. quasi-propositionality of the defaults (see Theorem 3.24)

2. universality (see Theorem 3.28)

Proof Overview: Non-trivial. Case (1.) follows immediately from Theorems 3.38 (immunity given "well-foundedness") and 3.24 ("well-foundedness" given quasi-propositionality). The proof of case (2.) is similar to the proof of Theorem 3.28 (satisfiability given universality). See Appendix. □

Proof of Theorem 3.39 :

Case (1.) Quasi-propositionality: was discussed in the proof overview.

Case (2.) Universality: By conservative extension for abnormalities (Theorem 3.8), it suffices to show the predicate case. Thus, by the proof of Theorem 3.38, it suffices to show condition (2) there, where in this case H is $(P; R; fix\ W)$:

$$\begin{aligned}&\forall W^1, Y^1.\ B[W^1, Y^1] \ \supset \\ &\quad [\exists W^2, Y^2.\ W^2 = W^1 \wedge PPC(B; P; R; fix\ W; W, Y)[W^2, Y^2]\,]\end{aligned} \tag{1}$$

Let $H2$ be defined as $(P; R2; fix\ W)$, where $R2$ is some total refinement of R, as in the proof of Theorem 3.28. By Corollary 4.3' of [Lifschitz, 1986], since $R2$ is layered, B is then "well-founded" with respect to $H2$. This implies condition (1) holds when $H2$ is substituted for H, i.e., when $R2$ is substituted for R:

[1] Recall page 36

$$\forall W^1, Y^1.\, B[W^1, Y^1] \;\supset$$
$$[\exists W^2, Y^2.\, W^2 = W^1 \wedge PPC(B; P; R2; fix\, W; W, Y)[W^2, Y^2]\,] \tag{2}$$

But, since $R \leq R2$, the monotonicity of prioritization (Theorem 2.58) implies that:

$$PPC(B; P; R2; fix\, W; W, Y) \;\models\; PPC(B; P; R; fix\, W; W, Y) \tag{3}$$

since, in the minimized prioritized pre-orders, we can regard the fixing of W as simply another starting pre-order that has empty prioritization relative to all the other starting pre-orders (each of which corresponds to the minimization of one predicate). (2) and (3) imply (1).

QED Theorem 3.39

Theorem 3.56 (Immunity of Fixed Relative)
Our earlier results showing immunity of the fixed Theorems 3.39 (universality or quasi-propositionality suffice) and 3.38 ("well-foundedness" suffices). generalize to **closed** (elementary) formulas that are fixed **relative** to the circumscription:

1. Let E be any closed formula. Then "well-foundedness" suffices for E to be immune. In particular, in PDC, quasi-propositionality suffices.

2. Let E be any quantifier-free formula. Then, in PDC, universality suffices for E to be immune.

Here, by immunity, we mean that:

$$C(B; H; Z) \models E[Z] \quad\iff\quad B[Z] \models E[Z]$$

Proof Overview: Not immediate. The proof for the "well-founded" case is similar to the proof of Theorem 3.38. By Theorem 3.24, quasi-propositionality implies "well-foundedness". For the universal case, we use Corollary 3.45. See Appendix for details. □

Proof of Theorem 3.56 :

"Well-founded" case: Let the circumscription be $C(B; H; Z)$. Examining the immunity condition for E, we see that the implication from right to left is easy: it follows from the fact that $C(B)$ always entails B. Thus, it suffices to show

$$[\forall Z.\, C(B; H; Z) \;\supset\; E(Z)] \;\supset\; [\forall Z.\, B(Z) \;\supset\; E(Z)] \tag{G}$$

The circumscription's "well-foundedness" is given:

$$\forall Z.\, B(Z) \;\supset\; \exists Z'.\; Z' \preceq_H Z \;\wedge\; C(B; H; Z)(Z') \tag{0.1}$$

Here, $C(B; H; Z)(Z')$ stands for $C(B; H; Z)$ with Z' substituted for Z'.

That E is fixed relative to the circumscription is also given:

$$\forall Z, Z'.\, B(Z) \wedge B(Z') \wedge Z' \preceq_H Z \;\supset\; (E(Z) = E(Z')) \tag{0.2}$$

A basic property of any circumscription is that it implies its base:

$$C(B;H;Z)(Z') \;\supset\; B(Z') \tag{1}$$

(0.1), (0.2), and (1) $\implies$

$$\forall Z.\; B(Z) \;\supset\; \exists Z'.\; (E(Z) = E(Z')) \;\wedge\; C(B;H;Z)(Z') \tag{2}$$

Using the given closed-ness of E: (2) $\iff$

$$\begin{aligned}&\forall Z.\; B(Z) \;\supset\\ &\qquad \exists Z'.\; [(E(Z) \wedge E(Z')) \;\vee\; (\neg E(Z) \wedge \neg E(Z'))] \;\wedge\; C(B;H;Z)(Z')\end{aligned} \tag{3}$$

$\implies$

$$\forall Z.\; B(Z) \;\supset\; \exists Z'.\; E(Z) \;\vee\; [\neg E(Z') \wedge C(B;H;Z)(Z')] \tag{4}$$

$\iff$

$$\forall Z.\; [\neg B(Z) \vee E(Z)] \;\vee\; [\exists Z'.\; C(B;H;Z)(Z') \wedge \neg E(Z')] \tag{5}$$

$\iff$

$$[\forall Z. \neg B(Z) \vee E(Z)] \;\vee\; [\exists Z'.\; C(B;H;Z)(Z') \wedge \neg E(Z')] \tag{6}$$

$\iff$

$$[\forall Z'. C(B;H;Z)(Z') \;\supset\; E(Z')] \;\supset\; [\forall Z. B(Z) \;\supset\; E(Z)] \tag{7}$$

$\iff$ (G). **QED "Well-founded" case**

Quasi-propositional case: By Theorem 3.24, quasi-propositionality implies "well-foundedness". Therefore, by the above, immunity holds.
QED quasi-propositional case

Universal case: Let $PDC(B;D;R;Z)$ be a PDC that is universal. By a PDC being universal, we mean in the sense of Theorem 3.28.

We want to show that

$$[\forall Z. PDC(B;D;R;Z) \;\supset\; E(Z)] \quad\equiv\quad [\forall Z. B(Z) \;\supset\; E(Z))] \tag{G}$$

Since E is fixed-relative, one can equivalently fix it as part of the maximized pre-order:

$$PDC(B;D;R;fix\ E;Z) \;\equiv\; PDC(B;D;R;Z) \tag{1}$$

Since E is quantifier-free, it is closed. Let par be a new 0-ary predicate symbol. (I.e., par does not appear in Z.) Corollary 3.45 $\implies$

$$\begin{aligned}&PDC(B \wedge (E(Z) \equiv par); D; R; fix\ par; Z, par)\\ &\qquad\equiv\quad (E(Z) \equiv par) \wedge PDC(B;D;R;fix\ E;Z)\end{aligned} \tag{2}$$

Since $E(Z)$ is quantifier-free, $E(Z) \equiv par$ is also quantifier-free. The left-hand-side PDC in (2) is thus universal. Therefore, the universal case of Theorem 3.39 implies that immunity holds for par in it:

$$PDC(B \wedge (E(Z) \equiv par); D; R; fix\ par; Z, par) \models par$$
$$\iff \quad B(Z) \wedge (E(Z) \equiv par) \models par \tag{3}$$

But, in (3), both antecedents also entail $E(Z) \equiv par$. Thus (3) $\implies$

$$PDC(B \wedge (E(Z) \equiv par); D; R; fix\ par; Z, par) \models E(Z)$$
$$\iff \quad B(Z) \wedge (E(Z) \equiv par) \models E(Z) \tag{4}$$

$\iff$

$$[\forall Z, par.\ PDC(B \wedge (E(Z) \equiv par); D; R; fix\ par; Z, par) \ \supset\ E(Z)]$$
$$\equiv \quad [\forall Z, par.\ B(Z) \wedge (E(Z) \equiv par) \ \supset\ E(Z)] \tag{5}$$

$\iff$

$$[\forall Z.\ (\exists par.\ PDC(B \wedge (E(Z) \equiv par); D; R; fix\ par; Z, par) \ \supset\ E(Z))]$$
$$\equiv \quad [\forall Z.\ (\exists par.\ B(Z) \wedge (E(Z) \equiv par) \ \supset\ E(Z))] \tag{6}$$

(1) and (2) imply that (6) $\iff$

$$[\forall Z.\ (\exists par.\ (E(Z) \equiv par) \wedge PDC(B; D; R; Z) \ \supset\ E(Z))]$$
$$\equiv \quad [\forall Z.\ (\exists par.\ (E(Z) \equiv par) \wedge B(Z) \ \supset\ E(Z))] \tag{7}$$

Lemma C.3 implies that (7) $\iff$ (G). **QED Universal case**

QED Theorem 3.56

C.4 For sub-section 3.5.6: Safety of Fixed Updates and Conclusions

Theorem 3.58 (Monotonicity of Fixed Base or Default Updates)
Suppose the sentence U is fixed relative to a circumscription. Then updating the base with U is globally monotonic: the circumscription after the update implies the circumscription before the update. More generally, suppose that the open formula U is fixed relative to a circumscription. Then updating with U as a default, with any priority, is also globally monotonic.

Proof Overview: Not terribly complicated. See Appendix. □

Proof of Theorem 3.58 :

Let the previous circumscription be $PDC(B; D; R; Z)$. (Recall that one can always view fixtures as polar defaults cf. Lemma 3.42.) Let E be a formula that is fixed relative:

$$\forall Z, Z'.\ B(Z) \wedge B(Z') \wedge Z' \preceq_{D;R} Z \ \supset\ E(Z') = E(Z) \tag{0}$$

Case I): Base Update: Notice that the circumscriptive policy is the same before and after the base update. We will show that

$$PDC(B \wedge E; D; R; Z) \;\equiv\; [E(Z) \;\wedge\; PDC(B; D; R; Z)] \tag{G1}$$

The left-hand-side of (G1) $\overset{\text{def}}{\iff}$

$$B(Z) \;\wedge\; E(Z) \;\wedge\; \neg\exists Z'.\, B(Z') \;\wedge\; E(Z') \;\wedge\; Z \prec_{D;R} Z' \tag{1}$$

(0) implies that (1) $\iff$

$$B(Z) \;\wedge\; E(Z) \;\wedge\; \neg\exists Z'.\, B(Z') \;\wedge\; E(Z) \;\wedge\; Z \prec_{D;R} Z' \tag{2}$$

$\iff$

$$B(Z) \;\wedge\; E(Z) \;\wedge\; \neg\exists Z'.\, B(Z') \;\wedge\; Z' \prec_{D;R} Z \tag{3}$$

$\overset{\text{def}}{\iff}$ the right-hand-side of (G1). **QED I)**

Case II): Default Update: Any default update can be serialized into two updates: a "first-part" update consisting of the single default axiom, and a "second-part" update consisting of zero or more prioritization axioms. By the monotonicity of prioritization (Theorem 2.58), the "second-part" update is globally monotonic. Therefore, it suffices to show that the "first part" update is globally monotonic. Next, we do so.

In the "first-part" update, the new default is in parallel with all of the previous defaults; the maximized global policy pre-order after this "first-part" update is:

$\preceq_{D;R} \;\wedge\; \preceq_E$

Therefore, by canonical conjunctive decomposition cf. Theorem 5.14, the circumscription after this "first-part" update is equivalent to:

$$PDC(B; D; R; guard\, E; Z) \;\wedge\; PDC(B; E; \emptyset; guard\, (D; R); Z) \tag{4}$$

The key insight is that guarding a fixed-relative formula has no effect.
In more detail:
Expanding, the left-hand-side of (4) $\iff$

$$B(Z) \;\wedge\; \neg\exists Z'.\, B(Z') \;\wedge\; Z \prec_{D;R} Z' \;\wedge\; E(Z) \leq E(Z') \tag{5}$$

The given fixed-relative condition (0) implies that (5) is equivalent to:

$$B(Z) \;\wedge\; \neg\exists Z'.\, B(Z') \;\wedge\; Z \prec_{D;R} Z' \tag{6}$$

But (6) $\overset{\text{def}}{\iff}$ $PDC(B; D; R; Z)$, the previous circumscription. Thus (4) logically implies the previous circumscription. **QED II**

QED Theorem 3.58

Appendix D

Proofs for Chapter 4

D.1 For sub-section 4.8.3: Rational Monotony Fails

Observation 4.23 (Rational Monotony Fails)
Rational monotony fails for even the parallel propositional predicate case of circumscription (which by Theorem 3.24 is "well-founded').

Proof Overview: In the Appendix: We give a counterexample, which involves only three defaults and no fixing.[1] $\Box$

Proof of Observation 4.23 : We first show a counterexample for the parallel propositional default case of circumscription. Then we extend it into a counterexample for the parallel propositional predicate case.

Default Case: Let the base language $\mathcal{L}$ be propositional. Let the tuple Y of primitive propositions in $\mathcal{L}$ be: $\langle p, q, r \rangle$. Let

$$B \stackrel{\text{def}}{\equiv} p \wedge (q \vee r)$$
$$E \stackrel{\text{def}}{\equiv} q$$
$$U \stackrel{\text{def}}{\equiv} r$$

Let there be three defaults, defined as:

$$D1 \stackrel{\text{def}}{\equiv} q \wedge r$$
$$D2 \stackrel{\text{def}}{\equiv} \neg(p \wedge \neg q \wedge r)$$
$$D3 \stackrel{\text{def}}{\equiv} \neg(p \wedge q \wedge r)$$

Let D stand for the tuple of these three. Let the prioritization partial order be empty, i.e., parallel.

$$R \equiv \emptyset$$

Let the policy be

$$pol \equiv (D; \emptyset; Y)$$

The globally maximized pre-order is thus:

$$\preceq_{D;\emptyset}$$

which is just equivalent to: $\leq D$

[1]This example was worked out jointly with Hector Geffner.

Next, we will show that rational monotony fails:

$$C(B; pol) \models E$$
$$C(B; pol) \not\models \neg U$$
$$C(B \wedge U; pol) \not\models E$$

To show this, we reason directly about models. Because the base language is propositional, each model is simply a truth assignment to the primitive propositions. There are exactly three models that satisfy B, i.e., there are three B-models:

$$M1 \stackrel{\text{def}}{\equiv} p \wedge q \wedge \neg r$$
$$M2 \stackrel{\text{def}}{\equiv} p \wedge \neg q \wedge r$$
$$M3 \stackrel{\text{def}}{\equiv} p \wedge q \wedge r$$

When a default is false (not satisfied) in a model, we say that it is *violated* there. Let Vi stand for the set of defaults that are violated in model Mi. Then

$$V1 = \{D1\}$$
$$V2 = \{D1, D2\}$$
$$V3 = \{D3\}$$

Consider the preferences among the B-models according to the globally maximized policy preorder. Notice that one model Mi is strictly preferred to another model Mj iff the set of defaults violated in Mi is a strict subset of those violated in Mj: i.e., iff $Vi \subset Vj$.

Thus, $M1$ is strictly preferred to $M2$, but there are no strict preferences between $M1$ and $M3$, nor between $M2$ and $M3$. Thus the set of maximally preferred B-models is $\{M1, M3\}$. Thus,

$$C(B; pol) \equiv (p \wedge q \wedge \neg r) \vee (p \wedge q \wedge r)$$

which is equivalent to $p \wedge q$. Therefore,

$$C(B; pol) \models E$$
$$C(B; pol) \not\models \neg U$$

Next, consider the models of $B \wedge U$. There are exactly two: $\{M2, M3\}$. As we discussed above, however, there is no strict preference between $M2$ and $M3$. Thus the set of maximally preferred models, among those satisfying $B \wedge U$, is also: $\{M2, M3\}$. Thus,

$$C(B \wedge U; pol) \equiv (p \wedge \neg q \wedge r) \vee (p \wedge q \wedge r)$$

which is equivalent to $p \wedge r$. Therefore,

$$C(B \wedge U; pol) \not\models E$$

QED Default case

Predicate Case: By conservative extension for abnormalities, Theorem 3.8, the default-case example above is easily extended into one for the predicate case. Introduce three additional primitive propositions into $\mathcal{L}$: $ab1, ab2, ab3$. Add

$$(\neg ab1 \supset D1) \wedge (\neg ab2 \supset D2) \wedge (\neg ab3 \supset D3)$$

to the base B. Let pol be to minimize the three abnormality predicates $ab1, ab2, ab3$ in parallel. Let E and U be as above. Then, as above,

$$C(B; pol) \models E$$
$$C(B; pol) \not\models \neg U$$

$$C(B \wedge U; pol) \quad \not\models \quad E$$

QED Default case

QED Observation 4.23

Appendix E

Proofs for Chapter 5

E.1 Generalization of Positivity / Negativity

In section B.10, we introduced the usual notion of syntactic positivity / negativity. Next, we generalize positivity / negativity in a manner that we will find useful for analyzing prioritized circumscriptions.

Definition E.1 (Generalized Positivity / Negativity)
1) with respect to a pre-order: Let $E(Z, x)$ be an open formula in a tuple of predicates Z, and a tuple of individual object variables x. We define E to be *positive with respect to a pre-order* H, where H is defined over Z, when:

$$\forall Z, Z'.\ Z \preceq_H Z' \ \supset\ E(Z) \leq E(Z') \tag{1}$$

Fact B.11 says that the usual notion of positivity then corresponds to the special case of positivity with respect to the pre-order $\preceq_{Q;\emptyset} \wedge \approx_Y$.

Selecting variables; notation: When we wish to indicate the variables to which H should be applied in E, we make them explicit. E.g., we might say that $E(Z)$ is positive *with respect to* $(\preceq_L; Q)$, where $Q \subseteq Z$. Part of the meaning of saying this is that all of $Z - Q$ are "held fixed". Let $Y \stackrel{\text{def}}{=} Z - Q$. Then $E(Z)$ is positive with respect to $(\preceq_L; Q)$ when:

$$\forall Z, Z'.\ Q \preceq_L Q' \wedge Y = Y' \ \supset\ E(Z) \leq E(Z')$$

which we can also write as:

$$\forall Q, Y, Y'.\ Q \preceq_L Q' \wedge Y = Y' \ \supset\ E(Q, Y) \leq E(Q', Y)$$

Negativity is defined in the same way as positivity, except that, above, the rightmost $\leq$'s, e.g., in (1), are replaced by $\geq$.

E.2 For section 5.3: Guard Reduction

Lemma E.2

Let H be a pre-order defined over tuples of (predicate) symbols similar to Z. Let $NEG_H(Z)$ be a formula that is negative with respect to $(\preceq_H; Z)$. (Here we are using our generalization of positivity / negativity from Definition E.1.) Let L be a tuple of formulas that are Z-free. Furthermore, let L be similar (Definition 2.4) to Z.
Then

$$[\exists Z.\, (L \preceq_H Z) \;\wedge\; NEG_H(Z)] \;\equiv\; NEG_H(L)$$

Proof : From right-to-left: substitute L for Z, and use the reflexivity of pre-orders. The left-to-right direction $\iff$

$$\forall Z.\, \{[L \preceq_H Z \;\wedge\; NEG_H(Z)] \;\supset\; NEG_H(L)\}$$

which follows from the given negativity condition. **QED**

Theorem 5.12 (Guard Reduction Method: General)
The guard(s) in any guarded circumscription can be replaced equivalently, in the sense of conservative extension, by: adding new predicates, fixing those predicates, and adding new base axiom(s) about them, as follows:

$$C(B; min\ H; guard\ G; Y) \;\equiv$$
$$\exists GP.\, C(B[Y] \wedge (Y \preceq_G GP); min\ H; fix\ GP; GP, Y)$$

Here the predicate tuple GP is similar to Y, and distinct from Y. H is applied to Y (not GP) in the right-hand-side, ordinary circumscription. Note that the resulting ordinary policy is always in the same expressive class as the guarded policy.

Proof Overview: Non-trivial. Somewhat similar to the conservative extension theorem for abnormality theories (Theorem 3.8). Uses a generalization of positivity / negativity. See Appendix. □

Proof of Theorem 5.12 :

$$\exists GP.\, C(B \wedge (Y \preceq_G GP); min\ H; fix\ GP;\ Y, GP) \tag{1}$$

$$\overset{\text{def}}{\iff}$$

$$\exists GP.\, Y \preceq_G GP \wedge B(Y) \wedge \neg\exists Y'.\, B(Y') \wedge Y' \prec_H Y \wedge Y' \preceq_G GP \tag{2}$$

In (2), consider the sub-expression consisting of everything to the right of $Y \preceq_G GP$; call it (2b). Next, we use our generalization of positivity / negativity from Definition E.1. It is straightforward to check that (2b) is negative with respect to $(\preceq_G; GP)$. Thus by Lemma E.2, (2) $\iff$

$$B(Y) \wedge \neg\exists Y'.\, B(Y') \wedge Y' \prec_H Y \wedge Y' \preceq_G Y \tag{3}$$

which is just the elaborated definition of $C(B; min\ H; guard\ G; Y)$.

QED Theorem 5.12

E.3 For section 5.4: Canonical Conjunctive Decomposition

Guide to Reader: The proof proper of Theorem 5.14 is on page 368.

Definition E.3 (Standard Form of a Guarded Policy)
Remember that:

- the fixture of symbols can be viewed as a fixture pre-order (recall sub-section 2.3.1, especially Theorem 2.19);
- any fixture pre-order can be viewed as a guard pre-order (or, alternatively, as part of the minimized pre-order) (recall Definition 5.6); and
- multiple guard pre-orders can be viewed as a single guard pre-order corresponding to their parallel composition (recall Definition 5.8).

Thus any guarded policy can be re-expressed in "standard form" as

$$(\mathit{min}\ H; \mathit{guard}\, G; \mathit{vary}\ Z)$$

where Z is the tuple of *all* symbols in the base language $\mathcal{L}$, and H and G are each a single pre-order defined over Z.

Definition E.4 (Policy Relation)
For a given guarded policy $\mathit{pol} \stackrel{\text{def}}{\equiv} (\mathit{min}\ H; \mathit{guard}\, G; \mathit{vary}\ Z)$, we define its corresponding (binary) *policy relation* POL as:

$$POL(Z', Z) \stackrel{\text{def}}{\equiv} Z' \prec_H Z \ \wedge\ Z' \preceq_G Z$$

or, a bit more abstractly, as

$$POL \stackrel{\text{def}}{\equiv} \prec_H \ \wedge\ \preceq_G$$

or

$$\prec_{pol} \stackrel{\text{def}}{\equiv} \prec_H \ \wedge\ \preceq_G$$

The policy relation is defined syntactically, similarly to a pre-order, as binary relation in the base language, and also model-theoretically.[1]

Lemma E.5 (Policy Relation is a Strict Partial Order)

[1] Given our assumption, on page33, that a model-theoretic correspondent exists for every pre-order of interest that we are discussing.

Any policy relation is a strict partial order. That is, POL is anti-reflexive and transitive.

Hence, we can write POL as $\prec_{pol}$, as we did in Definition E.4, without being misleading. However, see the comment about **abuse of notation** in Definition E.6.

Proof : Let *pol* and POL be as in Definition E.4. POL is anti-reflexive since $\prec_H$ is. POL is transitive since both $\prec_H$ and $\preceq_G$ are. **QED**

Definition E.6 (Non-Strict Version of the Policy Relation)

We will find it convenient to talk about the *non-strict version* of the policy relation (also called the *non-strict policy relation*):

$$\preceq_{pol} \stackrel{\text{def}}{\equiv} \preceq_H \wedge \preceq_G$$

where *pol* is $(min\ H; guard\ G)$.

$\preceq_{pol}$ is indeed a pre-order. Note that the usual transitivity properties hold between $\preceq_{pol}$ and $\prec_{pol}$.

Notational Caution: Nevertheless, this is somewhat of an **abuse of our notation**, since, for guarded policies, it is **not** the case (in general) that $\prec_{pol}$ is equivalent to $(\preceq_{pol} \wedge \neg\succeq_{pol})$. However, we find it convenient to do this anyhow. (We will not be using $\preceq_{pol}$ in such a way that this will cause trouble.)

Definition E.7 (Protection Policy Tactic POLF)

Let H be a tuple of starting pre-orders, each defined over a tuple Z. Let H be indexed by the tuple N, and let R be a prioritization partial order defined over N. Relative to the policy (pre-order) *pol* $\stackrel{\text{def}}{\equiv}$ $(min\ (H^N; R); vary\ Z)$, we define the **"protection policy"** $polf-i$ as:

$$(min\ Hi; fix\ H^{R_D(i)}; guard\ (H^{R_I(i)}; R^{R_I(i)}); vary\ Z)$$

which we can write more concisely as:

$$(min\ Hi; fix\ H^{D(i)}; guard\ (H; R)^{I(i)}; vary\ Z)$$

or, still more concisely, as:

$$(min\ i; fix\ D(i); guard\ R^{I(i)}; vary\ Z)$$

More formally, we **define** the **"protection policy tactic"** as the abstract function which maps any prioritized policy *pol* into the $polf-i$ as defined above.

$POLFi$, the policy relation (Definition E.4) corresponding to to $polf-i$, is:

$$POLFi(Z', Z) \stackrel{\text{def}}{\equiv} Z' \prec_{Hi} Z \ \wedge \ Z' \approx_{H^{D(i)}} Z \ \wedge \ Z' \preceq_{(H;R)^{I(i)}} Z$$

which we can write more concisely as:

$$\prec_{polf-i} \stackrel{\text{def}}{\equiv} \prec_i \wedge \approx_{D(i)} \wedge \preceq_{R^{I(i)}}$$

Lemma E.8

Let $S \subseteq N$ be up-closed with respect to R^N. Then

$$\preceq_{R^N:S} \;\equiv\; \preceq_{R^S}$$

Proof : S up-closed $\implies$

$$\forall i \in S.\;\; D(i) \cap S \;=\; D(i)$$

Then use the definitions of $\preceq_{R^N:S}$ and $\preceq_{R^S}$. **QED**

Lemma E.9

1. For all i, $D(i)$ is up-closed under R. I.e.,

$$\forall i, j.\;\; j \in D(i) \supset D(j) \subseteq D(i) \tag{1}$$

2. $\forall i, j.\;\; j \in I(i) \;\supset\; D(j) \subseteq DI(i)$ (2)

3. For all i, $DI(i)$ is up-closed.

$$\forall i.\;\; j \in DI(i) \supset D(j) \subseteq DI(i) \tag{3}$$

4. $\forall i, j.\;\; j \in D(i) \;\supset\; I(j) \subseteq DI(i)$ (4)

5. For all i, $(D(i) \cup \{i\})$ is up-closed.

$$\forall i, j. j \in (D(i) \cup \{i\}) \supset D(j) \subseteq (D(i) \cup \{i\}) \tag{5}$$

Proof : All of these properties follow, with a little massage, from the transitivity of R.

Lemma E.10

$$\forall i.\;\; \preceq_{R}D(i) \equiv \preceq_{R:D(i)} \tag{1}$$

$$\forall i.\;\; \preceq_{R}DI(i) \equiv \preceq_{R:DI(i)} \tag{2}$$

Proof : By Lemma E.8 and, respectively, (1) and (3) of Lemma E.9. **QED**

Lemma E.11

$$\approx_{D(i)} \;\supset\; [\,\preceq_{R^N:I(i)} \;\equiv\; \preceq_{R}I(i)\,] \tag{G}$$

Proof of Lemma E.11 : (G) $\iff$

$$\approx_{D(i)} \wedge \preceq_{R^N:I(i)} \;\equiv\; \approx_{D(i)} \wedge \preceq_{R}I(i) \tag{G2}$$

Now,

$$\approx_{D(i)} \supset \preceq_{R^N:D(i)} \tag{1}$$

since

$$\approx_{D(i)} \supset \forall k \in D(i).\ \preceq_k .$$

By (1), the left-hand-side of (G2) $\iff$

$$\approx_{D(i)} \wedge \preceq_{R^N:DI(i)} \tag{2}$$

Lemma E.10 implies that (2) is equivalent to the right-hand-side of (G2).

QED Lemma E.11

Lemma E.12

$$\prec_{R^N} \equiv \exists i \in N.\ \prec_{polf-i}$$

Note that N need not be finite.

Proof of Lemma E.12 : (Leaving N implicit for the remainder of the proof)

$\preceq_R \overset{\text{def}}{\iff} (\forall i. \approx_{D(i)} \supset \preceq_i)$

$\prec_R \overset{\text{def}}{\iff} \preceq_R \wedge \neg \succeq_R \iff$

$$(\forall i. \approx_{D(i)} \supset \preceq_i) \wedge \neg(\forall i. \approx_{D(i)} \supset \succeq_i) \tag{1}$$

$\iff$

$$(\forall i. \approx_{D(i)} \supset \preceq_i) \wedge (\exists i. \approx_{D(i)} \wedge \prec_i) \tag{2}$$

$\iff$

$$\exists i. \forall k. (\approx_{D(i)} \wedge \prec_i) \wedge (\approx_{D(k)} \supset \preceq_k) \tag{3}$$

Consider (3b), the quantifier-free part of (3), for the cases of k in terms of its prioritization relative to i. For $k = i$, (3b) is immediately equivalent to $\approx_{D(i)} \wedge \prec_i$. For k lower priority than i:

$$k \in DB(i) \implies i \in D(k) \implies (\prec_i \supset \neg \approx_{D(k)})$$

Thus in this case also, (3b) is equivalent to $\approx_{D(i)} \wedge \prec_i$. For k higher priority than i:

$$k \in D(i) \implies (\approx_{D(i)} \supset \approx_k)$$

Thus in this case, as well, (3b) is equivalent to $\approx_{D(i)} \wedge \prec_i$. The only remaining case is k incomparable to i; thus (3) $\iff$

$$\exists i. \approx_{D(i)} \wedge \prec_i \wedge [\forall k \in I(i). \approx_{D(k)} \supset \preceq_k]$$

$\iff$

$$\exists i.\ \prec_i \wedge \approx_{D(i)} \wedge \preceq_{R:I(i)}$$

which, using Lemma E.11, $\iff$

$$\exists i.\ \prec_i \wedge \approx_{D(i)} \wedge \preceq_{R^{I(i)}}$$

QED Lemma E.12

Theorem 5.14 (Conjunctive, Pre-Orders, with Guarding)
Any prioritized circumscription, or, more generally, any guarded prioritized circumscription, can be conjunctively decomposed into *guarded* circumscriptions in which a single of its starting pre-orders is minimized, as follows:

$$C(B; min\ (H^N; R); fix\ F; guard\ G; Z) \ \equiv$$

$$\bigwedge_{i \in N} C(B; min\ Hi; fix\ H^{R_D(i)}, F; guard\ (H^{R_I(i)}; R^{R_I(i)}), G; Z)$$

Proof Overview: Complicated. Uses several lemmas about prioritized pre-orders. See Appendix. $\Box$

Proof of Theorem 5.14 :

$$C(B; H^N; R; fix\ F; guard\ G; Z)$$

$\stackrel{\text{def}}{\iff}$

$$B(Z) \ \wedge \ \neg\exists Z'.\ B(Z') \ \wedge \ Z' \prec_{(H^N;R)} Z \ \wedge \ Z' \approx_F Z \ \wedge \ Z' \preceq_G Z$$

By Lemma E.12, $\iff$

$$B(Z) \ \wedge \ \neg\exists Z'.\ B(Z') \ \wedge \ [\bigvee_{i \in N} (Z' \prec_{polf-i} Z)] \ \wedge \ Z' \approx_F Z \ \wedge \ Z' \preceq_G Z$$

$\iff$

$$\bigwedge_{i \in N} [B(Z) \ \wedge \ \neg\exists Z'.\ B(Z') \ \wedge \ (Z' \prec_{polf-i} Z) \ \wedge \ Z' \approx_F Z \ \wedge \ Z' \preceq_G Z]$$

$\stackrel{\text{def}}{\iff}$ the right-hand-side of (G).
Note that N need not be finite.
QED Theorem 5.14

Lemma E.13

$$\prec_R \ \equiv \ \preceq_R \ \wedge \ \exists i.\ \prec_i \tag{G}$$

Proof of Lemma E.13 : From left to right in (G): Clearly, $\prec_R \supset \preceq_R$. By Lemma E.12, $\prec_R$ $\iff$

$$\exists i. \prec_{polf-i}$$

which, examining its definition, $\implies$

$$\exists i. \prec_i$$

From right to left in (G): Since $\prec_i \supset \neg\approx_i$,

$$\exists i. \prec_i \;\supset\; \exists i. \neg\approx_i$$

$\implies$

$$\exists i. \prec_i \;\supset\; \neg\approx_N$$

But by Lemma B.10, $\approx_N \equiv \approx_R$, thus

$$\exists i. \prec_i \;\supset\; \neg\approx_R$$

But $(\preceq_R \wedge \neg\approx_R) \equiv \prec_R$.
QED Lemma E.13

E.4 For section 5.5: Canonical Serial Decomposition

Guide to Reader: The proof proper of Theorem 5.32 (descending) is on page 377. The proof proper of Theorem 5.35 (descending) is on page 381.

Assumption for serial decomposition: N is Finite:
Note that, for all our results about serial decomposition in this dissertation, we **assume** that N is finite. This assumption is always met in CLD: it is always met in PDC where there are a finite number of default formulas.

Definition E.14 (Highest (Lowest) Priority Indices)
Let R be a prioritization defined over an index tuple N. Relative to R and N, we say that $i \in N$ is a *highest-priority* (*lowest-priority*) index iff there are no indices that have higher (lower) priority than i, i.e., iff $R_D(i) = \emptyset$ ($R_{DB}(i) = \emptyset$). We say that a *subset* of indices $E \subseteq N$ is highest (lowest) priority iff all its members are, and we say that E is *strictly* highest-priority (lowest-priority) iff all the remaining indices have lower (higher) priority than every index in E, i.e., iff $\forall i \in E, j \in N - E.\ R(i,j) \quad (R(j,i))$. Note that when there is no prioritization, then all indices are both highest- and lowest- priority, but none are strictly highest or lowest. Note also that if E is either strictly-highest-, or strictly-lowest- priority, then one can view it as corresponding to a module (when E indexes a set of prioritized-default pre-orders).

Definition E.15 (Highest Priority Among)

We say that an index i is *highest-priority among* a subset of indices $E \subseteq N$ when there are no higher-priority indices in E: $R_D(i) \cap E = \emptyset$. Note that, in general, there may be several highest-priority indices among any E.

Perspective:

The $POLFi$ "protection policy tactic" (Definition E.7) was for an *individual* module; while the following $POL{-}S$ is for a *group* of modules with a particular prioritization property.

Terminology:

An index subset that is up-closed we will also sometimes call *upper-priority*. An analogy is the sociological concept of "upper class".

Definition E.16 (Upper-Priority Group Protection Tactic $POL{-}S$)

Let $pol \overset{\text{def}}{\equiv} (min\ (H^N; R^N))$ be a (overall, i.e., global) policy. Let $S \subseteq N$ be up-closed (relative to R). Then we define $pol{-}S$, the group protection policy for S, relative to the (global) policy, as:

$$(min\ (H^S; R^S); guard\ (H^N; R^N))$$

which we can write a bit more concisely as

$$(min\ (H; R)^S; guard\ (H; R)^N; vary; Z)$$

or, still more concisely, as:

$$(min\ R^S; guard\ R^N)$$

More formally, we **define** the **"upper-priority group protection policy tactic"** as the abstract function which maps any prioritized policy pol into the $pol{-}S$ as defined above.

We define $POL{-}S$, the policy relation (Definition E.4) corresponding to $pol{-}S$, as:

$$POL{-}S(Z', Z) \overset{\text{def}}{\equiv} Z' \prec_{(H^S;R^S)} Z \quad \wedge \quad Z' \preceq_{(H^N;R^N)} Z$$

which we can write more concisely as:

$$\prec_{pol-S} \overset{\text{def}}{\equiv} \prec_{R^S} \quad \wedge \quad \preceq_{R^N}$$

Definition E.17 (Conjunction Operation on Policies)

We define

$$pol1\ \&\ pol2$$

to have the policy relation

$$\prec_{pol1} \vee \prec_{pol2}$$

Thus

$$C(B;(pol1\ \&\ pol2)) \equiv [C(B;pol1) \wedge C(B;pol2)]$$

We can re-state Lemma E.12, which underlies Theorem 5.14, using this notation, as:

$$\prec_{pol} \;\equiv\; \mathop{\&}_{i \in N} \prec_{polf-i}$$

Definition E.18 (Upper-Priority Sub-Theory)
Let $\mathcal{T}$ be (the circumscriptive theory corresponding to) $C(B;H^N;R;Z)$. Let $S \subseteq N$ be up-closed (relative to R and N). We define the *upper-priority sub-theory*, of $\mathcal{T}$, corresponding to the subset S, as (the circumscriptive theory corresponding to):

$$C(B;pol{-}S)$$

where $pol{-}S$ is defined as in Definition E.16.

As we will see shortly (in Lemmas E.21, E.22, E.25 and E.32):

For $S = \{i\}$, $pol{-}S \;\equiv\; polf{-}i$.

For $S = N$, $pol{-}S \;\equiv\; pol$. (Where pol is $(H^N;R;Z)$ here.)

$$pol{-}S \;\equiv\; \mathop{\&}_{i \in S} polf{-}i$$

$$pol{-}S \;\equiv\; \mathop{///}_{\substack{i \in S \\ via \\ descending\ sequence}} polf{-}i$$

Lemma E.19

$$\preceq_R \wedge \approx_{D(i)} \;\supset\; \preceq_{R}I(i)$$

Proof : $\preceq_R \implies \preceq_{R:I(i)}$. Then use Lemma E.11. **QED**

Lemma E.20

$$\prec_{polf-i} \;\equiv\; \prec_i \wedge \approx_{D(i)} \wedge \preceq_R$$

Proof : By Lemmas E.12 and E.19. **QED**

Lemma E.21
If $S = \{i\}$, then $pol{-}S \;\equiv\; polf{-}i$.

Proof : Since $\{i\}$ is up-closed, $D(i) = \emptyset$. Then use Lemma E.20. **QED**

Lemma E.22

For $S = N$, $pol\!-\!S \;\equiv\; pol$. In other words, $pol\!-\!N \;\equiv\; pol$.

Proof : Since $\prec_{R^N} \supset \preceq_{R^N}$, $\prec_{R^N} \equiv (\prec_{R^N} \wedge \preceq_{R^N})$. **QED**

Lemma E.23

Let $S \subseteq N$ be up-closed with respect to R^N. Then

$$\preceq_{R^N} \;\supset\; \preceq_{R^S}$$

Proof : $\preceq_{R^N} \implies \preceq_{R^N{}_{:S}}$. Then use Lemma E.8. **QED**

Lemma E.24

$$pol\!-\!(S \cup \{i\}) \;\equiv\; pol\!-\!S \;\&\; polf\!-\!i$$

Proof :

$$\preceq_{R^{S \cup \{i\}}} \;\equiv\; \preceq_{R^{S \cup \{i\}}{}_{:S}} \wedge [\approx_{D(i) \cap (S \cup \{i\})} \;\supset\; \preceq_i] \tag{1}$$

(The right-hand-side of the right-hand-side of (1) is just $\preceq_{R^{S \cup \{i\}}{}_{:\{i\}}}$.)

Since $S \cup \{i\}$ is up-closed,

$$D(i) \subseteq S \cup \{i\} \tag{2}$$

Since S is an up-closed subset of $S \cup \{i\}$, Lemma E.8 $\implies$

$$\preceq_{R^{S \cup \{i\}}{}_{:S}} \;\equiv\; \preceq_{R^S} \tag{3}$$

Thus, using the definition of $\prec$ as $(\preceq \;\wedge\; \neg\succeq)$, (1),(2),(3) $\implies$

$$\prec_{R^{S \cup \{i\}}} \;\equiv\; [\preceq_{R^S} \;\wedge\; (\approx_{D(i)} \supset \preceq_i)] \;\wedge\; [\neg\succeq_{R^S} \;\vee\; (\approx_{D(i)} \wedge \neg\succeq_i)] \tag{4}$$

The right-hand-side of (4) $\iff$

$$[\prec_{R^S} \;\wedge\; (\approx_{D(i)} \supset \preceq_i)] \;\vee\; [\preceq_{R^S} \;\wedge\; (\approx_{D(i)} \wedge \prec_i)]$$

Thus

$$\begin{aligned}\prec_{R^{S \cup \{i\}}} \wedge \preceq_{R^N} \;\equiv\; \{&[\prec_{R^S} \;\wedge\; \preceq_{R^N} \;\wedge\; (\approx_{D(i)} \supset \preceq_i)] \\ &\vee\; [\preceq_{R^S} \;\wedge\; \preceq_{R^N} \;\wedge\; (\approx_{D(i)} \wedge \prec_i)]\}\end{aligned}$$

However, $\preceq_{R^N}$ implies $\preceq_{R^N{}_{:\{i\}}}$, which is just equivalent to $\approx_{D(i)} \supset \preceq_i$. And, since S is up-closed, by Lemma E.23, $\preceq_{R^N}$ implies $\preceq_{R^S}$. Furthermore, by Lemma E.20,

$$\prec_i \wedge \preceq_{R^N} \wedge \approx_{D(i)} \;\equiv\; \prec_i \wedge \approx_{D(i)} \wedge \preceq_{R^{I(i)}}$$

Thus

$$\prec_{R^{S \cup \{i\}}} \wedge \preceq_{R^N} \;\equiv\; [\prec_{R^S} \;\wedge\; \preceq_{R^N}] \;\vee\; [\prec_i \wedge \approx_{D(i)} \wedge \preceq_{R^{I(i)}}]$$

which is just the elaborated statement of the lemma. **QED**

Lemma E.25

$$pol-S \;\equiv\; \mathop{\&}_{i \in S} \; polf-i$$

Proof : Our proof is by induction. We build up S by accumulating the indices according to a sequence that is descending (Definition 5.31) with respect to priority. The sequence starts with some "initial" index that is highest-priority. (One can always find such a sequence, since R^N is well-founded.) Base case: Since the "initial" S is a highest-priority singleton, use Lemma E.21. Inductive case: Use Lemma E.24. **QED**

Definition E.26 (Serial Policy)
We define serial policy in terms of series circumscription. Let $pol1$ and $pol2$ be two policies. We write

$$pol1 \;/\; pol2$$

to stand for the serial policy $spol$ formed by applying $pol1$ first, and then $pol2$.

$$C(B; spol) \;\stackrel{\text{def}}{\equiv}\; C(C(B; pol1); pol2)$$

$pol1$ and $pol2$ might each be a guarded policy, or, more generally, a serial policy. Note that $/$ is associative, but not, in general, commutative.

We define the empty (serial) policy $empty-policy$ as doing nothing:

$$C(B; empty-policy) \;\stackrel{\text{def}}{\equiv}\; B$$

Terminology: Policy Vs. Serial Policy:
Throughout this paper:
Unless it is clear otherwise in the context, when we simply say "policy", we refer to a policy that may be ordinary or guarded, but not serial.

Notation and Terminology: Sequences and Serial Policies:
We write $spol \stackrel{\text{def}}{\equiv} (tpol; \sigma)$ to indicate a serial policy, where $tpol$ is a tuple of policies (perhaps themselves serial policies), indexed by E, and the **policy sequence** σ is a sequencing of the indices of E, i.e., a permutation of E. We also write

$$(tpol; \sigma) \;\stackrel{\text{def}}{\equiv}\; \mathop{///}_{\substack{i \in E \\ via \\ \sigma}} \; tpol_i$$

to indicate that the policies in $tpol$ are to be applied in the sequence specified by σ:

$$tpol_{\sigma(1)} \;/\; tpol_{\sigma(2)} \;/\; \ldots$$

Let $\sigma[j:k]$ stand for the **sub-sequence** of σ from its j^{th} through its k^{th} members, inclusive:

$$\sigma[j:k] \overset{\text{def}}{\equiv} \langle \sigma(j), \sigma(j+1), \ldots, \sigma(k) \rangle$$

Then we define the corresponding **members** and **subsequences** of the serial policy *spol* as:

$$spol[j] \overset{\text{def}}{\equiv} tpol_{\sigma(j)}$$

$$spol[j:k] \overset{\text{def}}{\equiv} (spol[j] \ / \ spol[j+1] \ / \ \ldots spol[k])$$

We define $\sigma[1:0]$ to be **the empty sequence**. And we use $\sigma[j]$ as an alternative for $\sigma(j)$.

We write $(\sigma \ / \ \theta)$ to stand for the **concatenation** of the sequences σ and θ, i.e., σ followed by θ.

Lemma E.27 (Augmentation is Negative in the Policy Relation)

The augmentation of any (guarded) circumscription with policy $(min\ H; guard\ G)$ is is negative with respect to the pre-order $\preceq_H \wedge \preceq_G$, (i.e., its non-strict policy relation) and, therefore, with respect to its policy relation. I.e., for any base sentence B and policy $(min\ H; guard\ G)$:

$Z2 \preceq_H Z1 \ \wedge \ Z2 \preceq_G Z1 \ \models$
$[C_{aug}(B; min\ H; guard\ G; Z1) \ \supset \ C_{aug}(B; min\ H; guard\ G; Z2)]$

Proof : Follows easily from the transitivity of $\preceq_H$ and $\preceq_G$ when we examine the form of the C_{aug} minimization condition. **QED**

Proposition E.28 (Strengthening Base Weakens Augmentation)

Strengthening the base sentence of any (guarded) circumscription weakens its augmentation.

Proof of Proposition E.28 : Follows immediately from the form of the augmentation:

$$\neg \exists Z'. \ B(Z') \ \wedge \ Z' \prec_{pol} Z$$

QED Proposition E.28

Lemma E.29 (2-ary Serial Vs. Conjunctive: "Weaker" Policy First)

Let $\preceq_{H1}$, $\preceq_{H2}$, $\preceq_{G1}$, $\preceq_{G2}$ be pre-orders defined over Z. Suppose

$$\models \ \prec_{H2} \wedge \preceq_{G2} \ \supset \ \preceq_{H1} \wedge \preceq_{G1} \tag{0}$$

Then

$$(min \preceq_{H1}; guard \preceq_{G1}; Z) \ / \ (min \preceq_{H2}; guard \preceq_{G2}; Z)$$

is equivalent to

$$(min \preceq_{H1}; guard \preceq_{G1}; Z) \ \& \ (min \preceq_{H2}; guard \preceq_{G2}; Z)$$

More concisely: let $pol1$ and $pol2$ be two (guarded) policies such that (0) holds:

$$\models \ \prec_{pol2} \supset \preceq_{pol1}$$

In other words, let the non-strict version of $pol1$'s policy relation be weaker than $pol2$'s (strict) policy relation. Then

$$(pol1 \ / \ pol2) \ \equiv \ (pol1 \ \& \ pol2)$$

Proof :
Local Notation: we modify our usual notation for policies to make the variables (Z) explicit.
Let $B(Z)$ be an arbitrary base sentence. We want to show that

$$C(B; ((pol1; Z) \ / \ (pol2; Z))) \tag{1}$$

is equivalent to

$$C(B; ((pol1; Z) \ \& \ (pol2; Z))) \tag{2}$$

Let $B1(Z)$ be defined as $C_{aug}(B; pol1; Z)$. Then (1) $\iff$

$$B(Z) \ \wedge \ B1(Z) \ \wedge \ C_{aug}(B \wedge B1; pol2; Z) \tag{3}$$

and (2) $\iff$

$$B(Z) \ \wedge \ B1(Z) \ \wedge \ C_{aug}(B; pol2; Z) \tag{4}$$

Assume $B(Z) \ \wedge \ B1(Z)$. Then it suffices to show that (3b), the explicitly-notated C_{aug} in (3), is equivalent to (4b), the explicitly-notated C_{aug} in (4). By Proposition E.28, (4b) implies (3b). To show that (3b) implies (4b):
Comparing their elaborations:

$$\neg \exists Z'. \ B(Z') \wedge B1(Z') \wedge Z' \prec_{pol2} Z \tag{3b}$$

$$\neg \exists Z'. \ B(Z') \qquad\qquad \wedge Z' \prec_{pol2} Z \tag{4b}$$

it suffices to show

$$\forall Z'. \ Z' \prec_{pol2} Z \ \supset \ B1(Z') \tag{G}$$

By Lemma E.27, $B1$ is negative with respect to the pre-order $\preceq_{H1} \wedge \preceq_{G1}$, i.e., with respect to the non-strict version of the policy relation for $pol1$. But by (0), it must therefore be negative with respect to $\prec_{pol2}$. Thus our assumption $B1(Z)$ implies (G). **QED**

Lemma E.30

Let $pol1$ be $pol-S$, and $pol2$ be $polf-i$. Then condition (0) in Lemmas E.29, E.38, and E.39 is satisfied.

Proof : By Lemma E.12, $\prec_{polf-i}$ implies $\prec_{R^N}$, and thus $\preceq_{R^N}$.
By Lemma E.23, $\preceq_{R^N}$ implies $\preceq_{R^S}$. **QED**

Lemma E.31

$$pol-(S \cup \{i\}) \;\equiv\; (pol-S \;/\; polf-i)$$

Proof : Use lemmas E.29 and E.30. **QED**

Lemma E.32

Let σ be any sequencing of the indices in S that is descending (with respect to the prioritization R). Then

$$pol-S \;\equiv\; \underset{\substack{i \in S \\ via \\ \sigma}}{///} \; polf-i$$

Proof : Our proof is by induction. We build up S by accumulating the indices according to a sequence that is descending with respect to priority. The sequence starts with some "initial" index that is highest-priority. (One can always find such a sequence, since R^N is well-founded.) Base case: Since the "initial" S is a highest-priority singleton, use Lemma E.21.
Inductive case: Use Lemmas E.24 and E.31. **QED**

Lemma E.33

Let S be up-closed, and σ be any sequencing of S that is descending (with respect to the prioritization R). Then

$$\underset{i \in S}{\&} \; polf-i \;\equiv\; \underset{\substack{i \in S \\ via \\ \sigma}}{///} \; polf-i$$

Proof : Use Lemmas E.32 and E.25. **QED**

Lemma E.34

Let σ be any sequencing of N that is descending (with respect to the prioritization R). Then

$$pol-N \;\equiv\; \underset{\substack{i \in N \\ via \\ \sigma}}{///} \; polf-i$$

Proof : Use Lemmas E.32 and E.22. **QED**

Theorem 5.32 (Serial, Descending, General Priority: Pre-Orders)
Let σ be a permutation of the index set N that is descending with respect to the prioritization partial order R. Let N be finite. Then

$$C(B; (H^N; R); Z) \quad \equiv$$

$$\underset{\substack{i \in N \\ via \\ \sigma}}{///} \; C(\underset{B}{\bullet} ; H_i; fix\ H^{R_D(i)}; guard\ (H^{R_I(i)}; R^{R_I(i)}); Z)$$

Proof Overview: Complicated. By well-founded induction on the prioritization partial order. Uses Theorem 5.14 plus lemmas about properties of prioritized pre-orders. See Appendix. □

Proof of Theorem 5.32 : Lemma E.34.
QED Theorem 5.32

Definition E.35 (Well-Foundedness: Guarded Policies)
We say the base sentence B is well-founded with respect to the *guarded* policy *pol* when there does not exist an infinite descending chain of models in which each model satisfies B and each model is $\prec_{pol}$ than the previous model in the chain.

Lemma E.36 (Guarding Preserves Well-Foundedness)
Adding guards preserves well-foundedness.
More precisely: Suppose a base sentence B is well-founded with respect to the policy $pol1$. Then B is well-founded with respect to the policy $pol2$, where:

$$\prec_{pol2} \quad \equiv \quad \prec_{pol1} \wedge \preceq_G$$

where G is any (guard) pre-order.

In particular, if B is well-founded with respect to the policy $(min\ H)$, then B is well-founded with respect to the policy $(min\ H; fix\ F; guard\ G)$.

Proof : Since

$$\models \quad \prec_{pol2} \supset \prec_{pol1}$$

it follows that any infinite chain of B-models that is decreasing with respect to $\prec_{pol2}$ must also be an infinite chain of B-models that is decreasing with respect to $\prec_{pol1}$. **QED**

Definition E.37 (Well-Foundedness: Serial Policies)
We say that the base sentence B is well-founded with respect to the *serial* policy

$$spol \overset{\text{def}}{\equiv} (tpol; \sigma)$$

when

$$C(B; spol[1:j-1]) \text{ is well-founded with respect to } spol[j]$$

for each $j \geq 1$. I.e., when the tier corresponding to each prefix of the serial policy is well-founded with respect to the next member of the serial policy.
(Note that well-foundedness with respect to the empty (serial) policy is trivially true.)

Lemma E.38 (2-ary Serial Vs. Conjunctive: "Weaker" Second)
Let $pol1$ and $pol2$ be two guarded policies, defined over Z, such that

$$\models \ \prec_{pol2} \ \supset \ \preceq_{pol1} \tag{0}$$

(as in Lemma E.29).
If, in addition,

the base sentence B is well-founded (Definition E.35) with respect to $pol2$, (W)

then (in the context of B):

$$(pol2 \ / \ pol1) \ \equiv \ (pol2 \ \& \ pol1)$$

Proof : The first part of the proof is similar to the proof of Lemma E.29, but with $1 \leftrightarrow 2$. Let

$$B2(Z) \ \stackrel{\text{def}}{\equiv} \ C_{aug}(B; pol2; Z)$$

$$B3(Z) \ \stackrel{\text{def}}{\equiv} \ C_{aug}(B \wedge B2; pol1; Z)$$

$$B1(Z) \ \stackrel{\text{def}}{\equiv} \ C_{aug}(B; pol1; Z)$$

Assume $B(Z)$ and $B2(Z)$ (A1)

Then it suffices to show that $B3(Z) \equiv B1(Z)$. But $B1(Z)$ implies $B3(Z)$ by Proposition E.28.

To show that $B3(Z)$ implies $B1(Z)$: By contradiction. Assume $B3(Z)$ and $\neg B1(Z)$. Elaborating the definitions of our assumptions:
$B3(Z) \ \stackrel{\text{def}}{\Longleftrightarrow}$

$$\neg\exists Z'. \ B(Z') \ \wedge \ B2(Z') \ \wedge \ Z' \prec_{pol1} Z \tag{2}$$

$\neg B1(Z) \ \stackrel{\text{def}}{\Longleftrightarrow}$

$$\exists Y_1. \ B(Y_1) \ \wedge \ Y_1 \prec_{pol1} Z \tag{3.0}$$

The rest of the proof is, until the last line or two, within the scope of (3.0): that is, we accumulate conclusions to append to (3.0), within the scope of its quantifier(s).
The quantifier-free part of (3.0), and (2) with Y_1 substituted for Z', $\Longrightarrow$

$$\neg B2(Y_1) \tag{3.1}$$

$B2(Y_1) \ \stackrel{\text{def}}{\Longleftrightarrow}$

$$\neg\exists Z'.\ B(Z')\ \wedge\ Z' \prec_{pol2} Z \tag{DEFN-B2}$$

(DEFN-B2) implies that (3.1) $\iff$

$$\exists Y_2.\ B(Y_2)\ \wedge\ Y_2 \prec_{pol2} Y_1 \tag{3.2}$$

The right-hand-side of (3.2), (0) $\implies$

$$Y_2 \preceq_{pol1} Y_1 \tag{3.3}$$

By transitivity of the minimization and guard pre-orders in $pol1$:

$$\forall U, V, W.\ U\ \preceq_{pol1} V\ \wedge\ V \prec_{pol1} W\ \ \supset\ \ U \prec_{pol1} W \tag{TRANS-POL1}$$

(3.3), the right-hand-side of (3.0), (TRANS-POL1) $\implies$

$$Y_2 \prec_{pol1} Z \tag{3.4}$$

The left-hand-side of (3.2), (3.4) $\implies$

$$\exists Y_2.\ B(Y_2)\ \wedge\ Y_2 \prec_{pol1} Z \tag{3.5}$$

(Note the similarity of (3.5) to (3.0).)
(3.5), and (2) with Y_2 substituted for Z', $\implies$

$$\neg B2(Y_2) \tag{3.6}$$

We have reached a "loop" in the proof. Essentially, (3.6) is the same "place" as (3.1). We now show another iteration.
(3.6), (DEFN-B2) $\implies$

$$\exists Y_3.\ B(Y_3)\ \wedge\ Y_3 \prec_{pol2} Y_2 \tag{3.7}$$

The right-hand-side of (3.7), (0) $\implies$

$$Y_3 \preceq_{pol1} Y_2 \tag{3.8}$$

(3.4), (3.8), (TRANS-POL1) $\implies$

$$Y_3 \prec_{pol1} Z \tag{3.9}$$

The left-hand-side of (3.7), (3.9) $\implies$

$$\exists Y_3.\ B(Y_3)\ \wedge\ Y_3 \prec_{pol1} Z \tag{3.10}$$

(3.10), and (2) with Y_3 substituted for Z', $\implies$

$$\neg B2(Y_3) \tag{3.11}$$

Steps (3.7) to (3.11) are isomorphic to steps (3.2) to (3.6) Clearly, we can continue in this way forever, in each iteration concluding the existence of a new Y_{k+1}. Note that each Y_{k+1} has the property, shown above in the right-hand-side of (3.2) (and (3.7)), that

$$Y_{k+1} \prec_{pol2} Y_k$$

as well as the property, shown above in the left-hand-side of (3.5) (and (3.10)), that

$$B(Y_{k+1})$$

Note also that $B(Y_1)$ holds, by the left-hand-side of (3.0).

Thus, we have shown that there exists an infinite chain of Y_k's, $k = 1, 2, 3, \ldots$, such that for each k:

$$Y_{k+1} \prec_{pol2} Y_k \;\; \wedge \;\; B(Y_{k+1}) \;\wedge\; B(Y_k)$$

But this contradicts the well-foundedness of B with respect to $pol2$, given in (W). **QED**

Alternatively, in the proof of Lemma E.38, one could think of the chain model-theoretically.

Comment on Lemma E.38:
We know of no other result about circumscription that depends on well-foundedness as opposed to "well-foundedness".

Lemma E.39 (Serial Flip, Given Well-Foundedness)
Let $pol1$ and $pol2$ be two guarded policies, defined over Z, such that

$$\models \;\; \prec_{pol2} \;\supset\; \preceq_{pol1} \tag{0}$$

If, in addition,

the base sentence B is well-founded (Definition E.35) with respect to $pol2$, (W)

then (in the context of B) the sequencing of the two policies is irrelevant:

$$(pol2 \,/\, pol1) \;\; \equiv \;\; (pol1 \,/\, pol2)$$

and both serial policies are equivalent to

$$(pol1 \;\&\; pol2)$$

Proof : Use Lemmas E.29 and E.38. **QED**

Theorem 5.35 (Serial, Any Sequence, General Priority: Pre-Orders)
In Theorem 5.32, suppose that B is well-founded (Definition 3.18) with respect to every serial policy that corresponds to any sequence of the phase policies. (See the proof for the definition of such well-foundedness with respect to serial policies.) Quasi-propositionality (Theorem 3.24) of a global PDC, for example, suffices to ensure such well-foundedness in the prioritized-default case. (Note that the condition is the usual well-foundedness, *not* "well-foundedness".)

Then the sequencing σ may be *any* permutation of the index set N.

Proof Overview: Complicated. A nested induction on permutation of the sequences, using a central intuition of a vacuum cleaner or snowball (seriously!). Uses Theorem 5.32 and lemmas for the 2-ary case. See Appendix. □

Proof of Theorem 5.35 :

By the "main part" of the Theorem, we mean the serial decomposition given serial well-foundedness. The other part of the Theorem is that quasi-propositionality ensures sufficient well-foundedness.

Main part of the Theorem:

We can re-state the main part of the theorem as follows.

Let σ be an arbitrary permutation of N. Let n be the size of N (remember, we assumed that N is finite). Let the base sentence B be well-founded with respect to every serial policy corresponding to a subsequence of σ, i.e., with respect to every $polf-\delta$ (see $polfin\pi$ notation below), where δ is a subsequence of σ. (See Definition E.37 of serial well-foundedness.) Then

$$C(B; polf-\sigma) \;\equiv\; C(B; pol)$$

Notation:

For any sequence π, we write $polf-\pi$ to stand for

$$polf-\pi(1) \;/\; polf-\pi(2) \;/\; \ldots$$

Synopsis of Proof:

We make a somewhat unusual style of inductive argument. We show how to permute the policy sequence σ, via equivalence-transforming transformation operations, into a policy sequence that is descending with respect to priority. (We are speaking here of equivalence of the serial policies in the context of the base B.) The basic transformation operation is the pairwise exchange, i.e., *flip*, of a sub-sequence and an individual member. Finding such a path of flips is a bit tricky. Not all flips maintain equivalence. The essential intuition is of a *vacuum cleaner* or *snowball*: we create a growing sub-sequence that is descending and whose index set is up-closed, moving the sub-sequence up and down the sequence to find new indices to add to itself. Lemma E.39, along with Lemmas E.32 and E.30, is exploited to sanction the motion along the sequence: together they say that an up-closed descending subsequence can, equivalently, be moved past a member of the sequence by flipping with it.

Body of Proof:

The (top-level) **induction** is on the parameter j of an ordered-ness property of policy sequences α_j that are permutations of σ, i.e., permutations of N. Each $polf-\alpha_j$ will represent a serial policy that is equivalent to $polf-\sigma$ (in the context of B):

$$C(B; polf-\alpha_j) \;\equiv\; C(B; polf-\sigma)$$

Each α_j will be defined as the result of applying flip operations to its predecessor α_{j-1}. We start off with $\alpha_1 \overset{\text{def}}{\equiv} \sigma$. We end up with an α_n that is descending. The orderedness property is the presence in α_j of a descending subsequence μ_j whose indices form a set S_j, of size j, that is up-closed. (By "subsequence", we mean a consecutive subsequence.)

Our overall (top-level) induction parameter, then, is the size j of this subsequence μ_j of α_j. The (top-level) inductive hypothesis (H) is defined as the conjunction of (H1) and (H2):

$$C(B; polf-\alpha_j) \equiv C(B; polf-\sigma) \tag{H1}$$

$$\alpha_j \text{ is a permutation of } N \text{ and has a subsequence } \mu_j \text{ such that} \tag{H2.0}$$

$$S_j \text{ is of size } j \tag{H2.1}$$

$$S_j \text{ is up-closed} \tag{H2.2}$$

$$\mu_j \text{ is descending} \tag{H2.3}$$

(We define (H2) as the conjunction of (H2.0) through (H2.3).)
When j grows to be n, the length of the whole sequence, then α_j is exactly μ_j and thus is descending. (Remember, n is the size of N.) Each step of the top-level induction will itself be proved by a second-level induction. The final step of the overall proof comes when we use Theorem 5.32 to show that

$$C(B; polf-\alpha_n) \equiv C(B; pol)$$

Base Case of the inductive hypothesis (H): $j = 1$.
Let $\alpha_1 \stackrel{\text{def}}{=} \sigma$.
By the well-foundedness of the prioritization partial order R, and since σ is a permutation of N, there must be at least one highest-priority (Definition E.14) index; i.e., there must be at least one index $k_1 \in N$, appearing in σ, such that $R_D(k_1) = \emptyset$. Choose one such k_1 and let μ_1 be defined as the singleton sequence $\langle k_1 \rangle$. Then μ_1's index set is just the singleton set $\{k_1\}$ which is of size 1, is up-closed, and is descending.
QED Base Case

If n=1, then we are done. For the rest of the proof, suppose $n > 1$.
Inductive Case (top-level):
Assume (H) for j, where $1 \leq j < n$. We aim to show (H) for $j + 1$. Our method is to show that there exists a next sequence α_{j+1} satisfying (H).

Consider α_j. We can write it as

$$\varphi_j \;/\; \mu_j \;/\; \theta_j$$

where one, but not both, of the subsequences φ_j and θ_j may be empty. Consider the indices appearing in φ_j / θ_j: this set is just $N - S_j$. Since the prioritization (R) is well-founded, there must be at least one index that is highest-priority among $N - S_j$, i.e., there must be at least one index $k_{j+1} \in (N - Sj)$ such that

$$(R_D(k_{j+1}) \cap (N - S_j)) = \emptyset$$

$\Longleftrightarrow$

$$R_D(k_{j+1}) \subseteq S_j \tag{1}$$

We define μ_{j+1} to be μ_j / $\langle k_{j+1}\rangle$. Then S_{j+1} is $S_j \cup \{k_{j+1}\})$. Since S_j is up-closed and of size j, S_{j+1} is thus of size $j+1$, and by (1), is up-closed. The condition that μ_{j+1} be descending, given that μ_j is descending, is equivalent to:

$\forall g \in Sj.\ k_{j+1} \notin R_D(g)$

Since S_j is up-closed, in addition to μ_j being descending, μ_{j+1} thus is descending. Therefore, condition (H2) is satisified for $j+1$. We next show that there exists an α_{j+1}, a permutation of N, containing sub-sequence μ_{j+1}, such that condition (H1) is satisfied.

Either k_{j+1} appears before μ_j in α_j, or it appears after it. That is: either **I)** $k_{j+1} \in \varphi_j$ or **II)** $k_{j+1} \in \theta_j$. The two cases are similar.

I): Consider the case where $k_{j+1} \in \varphi_j$. We can write φ_j as

$$\eta_j \ / \ \langle k_{j+1}\rangle \ / \ \psi_j$$

where η_j and ψ_j each may be empty.

We define α_{j+1} as the result of "moving" the sub-sequence μ_j, within α_j, backwards in the sequence until it immediately precedes k_{j+1}:

$$\alpha_{j+1} \ \stackrel{\text{def}}{=} \ \eta_j \ / \ \mu_j \ / \ \langle k_{j+1}\rangle \ / \ \psi_j \ / \ \theta_j$$

Sub-Proof:

We now show that (H1) holds for $j+1$:

$$C(B; polf - \alpha_{j+1}) \ \equiv \ C(B; polf - \sigma) \tag{GG1}$$

By our assumption of (H1) for j, it suffices to show :

$$C(B; polf - \alpha_{j+1}) \ \equiv \ C(B; polf - \alpha_j) \tag{GG2}$$

We show this by (a second-level) **induction**.

The basic idea is that we can perform a succession of flips in order to "move" μ_j "through" $\langle k_{j+1}\rangle$ / ψ_j, past one index per flip, while at the same time maintaining equivalence of the resulting intermediate sequences to α_j. Let m be defined as the length of $\langle k_{j+1}\rangle$ / ψ_j. Since the set of indices appearing in $\langle k_{j+1}\rangle$ / ψ_j is a subset of $N - S_j$,

$$1 \leq m \leq (n-j)$$

Our (second-level) inductive parameter h will represent the number of steps backward we have taken in moving μ_j from its "initial" position after ψ_j (when we start with sequence α_j). Let

$$\rho_j \ \stackrel{\text{def}}{=} \ \langle k_{j+1}\rangle \ / \ \psi_j$$

Recalling our notation for sub-sequences: let

$\rho_j[l] \ \stackrel{\text{def}}{=}$ the l^{th} index in the sequence ρ_j

$\rho_j[1:l] \stackrel{\text{def}}{=}$ the initial subsequence of ρ_j up through its l^{th} position

$\rho_j[l+1:m] \stackrel{\text{def}}{=}$ the final subsequence of ρ_j starting at its $(l+1)^{th}$ position

Let α_{jh} stand for the intermediate sequence resulting after moving μ_j h steps backwards, starting from α_j; i.e., after moving μ_j backwards past h indices:

$$\alpha_{jh} \stackrel{\text{def}}{=} \eta_j \;/\; \rho_j[1:m-h] \;/\; \mu_j \;/\; \rho_j[m-h+1:m] \;/\; \theta_j$$

so that:

$$\alpha_{j0} = \alpha_j \tag{0.1}$$

$$\alpha_{jm} = \alpha_{j+1} \tag{0.2}$$

Our (second-level) inductive hypothesis is thus:

$$C(B; polf-\alpha_{jh}) \equiv C(B; polf-\alpha_j) \tag{GG3}$$

(GG3) for $h = m$ is, by (0.2), just (GG2). Thus, showing the second-level induction is sufficient to show (GG1).

Base case of sub-proof: $h = 0$: trivial: follows from (0.1).

Inductive (second-level) case:

Assume (GG3) for h. We aim to show (GG3) for $h+1$.

By (GG3) for h, it suffices to show :

$$C(B; polf-\alpha_{jh+1}) \equiv C(B; polf-\alpha_{jh}) \tag{GG4}$$

Comparing the two sequences:

$$\alpha_{jh+1} = \xi_{jh} \;/\; \mu_j \;/\; \langle\rho_j[m-h]\rangle \;/\; \nu_{jh}$$

$$\alpha_{jh} = \xi_{jh} \;/\; \langle\rho_j[m-h]\rangle \;/\; \mu_j \;/\; \nu_{jh}$$

where we write

$$\xi_{jh} \stackrel{\text{def}}{=} \eta_j \;/\; \rho_j[1:m-h-1]$$

$$\nu_{jh} \stackrel{\text{def}}{=} \rho_j[m-h+1:m] \;/\; \theta_j$$

Using the definition of series circumscription:
the left-hand-side of (GG4) $\Longleftrightarrow$

$$C(\; C(\; C(B; polf-\xi_{jh}) \;;\; polf-(\mu_j \;/\; \langle\rho_j[m-h]\rangle)) \;;\; polf-\nu_{jh}) \tag{1}$$

while the right-hand-side of (GG4) $\Longleftrightarrow$

$$C(\; C(\; C(B; polf-\xi_{jh}) \;;\; polf-(\langle\rho_j[m-h]\rangle \;/\; \mu_j)) \;;\; polf-\nu_{jh}) \tag{2}$$

Let

$$BB \stackrel{\text{def}}{\equiv} C(B; polf - \xi_{jh})$$

Comparing (1) and (2) implies that it suffices to show that

$$C(BB; polf - (\mu_j \ / \ \langle \rho_j[m-h] \rangle)) \tag{3}$$

is equivalent to

$$C(BB; polf - (\langle \rho_j[m-h] \rangle \ / \ \mu_j)) \tag{4}$$

Again using the definition of series circumscription:
(3) $\iff$

$$C(\ C(BB; polf - \mu_j) \ ; \ polf - \langle \rho_j[m-h] \rangle) \tag{5}$$

and (4) $\iff$

$$C(\ C(BB; polf - \langle \rho_j[m-h] \rangle) \ ; \ polf - \mu_j) \tag{6}$$

By (H2) for j, μ_j is a descending sequencing of the up-closed index set S_j. Thus by Lemma E.32:

$$polf - \mu_j \ \equiv \ pol\text{–}S_j \tag{7}$$

in the context of any base sentence. (7) implies that (5) $\iff$

$$C(\ C(BB; pol\text{–}S_j) \ ; \ polf - \langle \rho_j[m-h] \rangle) \tag{8}$$

and that (6) $\iff$

$$C(\ C(BB; polf - \langle \rho_j[m-h] \rangle) \ ; \ pol\text{–}S_j) \tag{9}$$

We next show that Lemma E.39 can be applied to flip $polf - \langle \rho_j[m-h] \rangle$ and $pol\text{–}S_j$. Remember that $\rho_j[m-h]$ is just a single index. Thus, by Lemma E.30, the condition (0) in Lemma E.39 is satisfied, for the choice of $polf - \langle \rho_j[m-h] \rangle$ as $pol2$, and $pol\text{–}S_j$ as $pol1$. By our well-foundedness assumption (W), BB is well-founded with respect to $polf - \rho_j[m-h]$. Thus the conditions for applying Lemma E.39 are satisified, and, hence, using it, (8) $\iff$ (9).
QED Inductive (second-level) case of sub-proof
QED Sub-proof

We have thus shown for the case (**I)**) where k_{j+1} precedes μ_j that there exists a suitable α_{j+1}.
QED I)

II): Now, let's consider the case (**II)**) where k_{j+1} follows μ_j, i.e., $k_{j+1} \in \theta_j$. The proof for this case is quite similar to that for case **I)**. Instead of moving μ_j backwards, we are moving it forwards. However, the key step in the proof of case **I)**, namely the ability to flip equivalently, is symmetric with respect to foward versus backward direction, and thus the validity of the proof does not depend on the direction of moving μ_j.

In detail: in the proof of case **I)**, swap the roles of φ_j and θ_j. Write θ_j as

$$\psi_j \ / \ \langle k_{j+1} \rangle \ / \ \eta_j$$

Define

$$\alpha_{j+1} \ \stackrel{\text{def}}{=} \ \varphi_j \ / \ \psi_j \ / \ \mu_j \ / \ \langle k_{j+1} \rangle \ / \ \eta_j$$

There is a slight difference from the backwards direction in that we move μ_j forward through ψ_j only, rather than through $\psi_j \ / \ \langle k_{j+1} \rangle$.

Sub-proof:

Let m be the length of ψ_j. Then

$$0 \le m \le (n - j - 1)$$

Let

$$\rho_j \ \stackrel{\text{def}}{=} \ \psi_j$$

$$\alpha_{jh} \ \stackrel{\text{def}}{=} \ \varphi_j \ / \ \rho_j[1:h] \ / \ \mu_j \ / \ \rho_j[h+1:m] \ / \ \langle k_{j+1} \rangle \ / \ \eta_j$$

If $m = 0$, we are done.

Base case of (second-level) induction: $h = 0$: trivial, since $\alpha_{j0} = \alpha_j$.

Inductive case of (second-level) induction:

$$\alpha_{jh+1} \ = \ \xi_{jh} \ / \ \mu_j \ / \ \langle \rho_j[h+1] \rangle \ / \ \nu_{jh}$$

$$\alpha_{jh} \ = \ \xi_{jh} \ / \ \langle \rho_j[h+1] \rangle \ / \ \mu_j \ / \ \nu_{jh}$$

where we write

$$\xi_{jh} \ \stackrel{\text{def}}{=} \ \varphi_j \ / \ \rho_j[1:h]$$

$$\nu_{jh} \ \stackrel{\text{def}}{=} \ \rho_j[h+2:m] \ / \ \langle k_{j+1} \rangle \ / \ \eta_j$$

Then the rest of the proof for case **II)** is as for case **II)**.

QED II)

Thus we have shown that whether μ_j is moved forwards or backwards after the choice of k_{j+1}, that (H) holds for $j+1$.

QED Inductive Case (top-level)

We have proved our top-level inductive hypothesis. Thus (H) holds for n. In particular, (H1) holds for n:

$$C(B; polf - \sigma) \ \equiv \ C(B; polf - \alpha_n) \tag{2}$$

But α_n is equal to μ_n, which, by (H2) for n, is descending. Thus, by Theorem 5.32, the right-hand-side of (2) is equivalent to $C(B; pol)$. Thus, we have shown that

$$C(B; polf - \sigma) \equiv C(B; pol)$$

QED Main part of the Theorem

Quasi-propositionality implies sufficient serial well-foundedness: Suppose the circumscription is a quasi-propositional PDC:

$PDC(B; D; R; Z)$

We observe first that the definition of quasi-propositionality, Definition 3.22, generalizes straightforwardly to guarded PDC's: i.e., to permitting an arbitary pre-order guard to be present. Then Lemma 3.23 (quasi-propositionality implies each open default is equivalent to its instances) generalizes straightforwardly as well. By Lemma E.36 (guarding preserves well-foundedness), Theorem 3.24 thus generalizes straightforwardly to guarded PDC's, in that: quasi-propositionality implies well-foundedness with respect to the guarded policy.

Therefore, it suffices to show each phase in the serial decomposition (in the main part of our Theorem) is quasi-propositional.

By the definition of quasi-propositionality (Definition 3.22), all function symbols W in Z, that appear in the default formulas D, are fixed relative (Definition 3.47) to the global circumscription:

$$\forall Z, Z'.\ B[Z] \wedge B[Z'] \wedge Z \preceq_{D;R} Z' \ \supset\ W = W' \tag{0}$$

Consider any phase: it has some base BPg, and some guarded phase policy $polf - g$ in which the single default Dg is maximized:

$$MC(BPg; polf - g; Z) \tag{Phase-g}$$

where

$$polf - g \ \stackrel{\text{def}}{=}\ (max\ Dg; fix\ D^{R_D(g)}; guard\ (D; R)^{R_I(g)})$$

(Recall that MC stands for "Maximizing Circumscription".)

The phase base BPg is the tier resulting from the previous phase. Since each previous phase circumscription implies its own base, BPg thus entails the initial base B of the series:

$$BPg[Z] \ \models\ B[Z] \tag{1}$$

Lemma E.20 implies that

$$\models \quad \prec_{polf - g} \supset \preceq_{D;R} \tag{2}$$

(0), (1), and (2) $\Longrightarrow$

$$\forall Z, Z'.\ BPg[Z] \wedge BPg[Z'] \wedge Z \prec_{polf - g} Z' \ \supset\ W = W' \tag{3}$$

Let $polgw$ be defined as the adding of the fixture of W to $polf-g$:

$$polgw \stackrel{\text{def}}{\equiv} (max\ Dg; fix\ D^{R_D(g)}, W; guard\ (D; R)^{R_I(g)})$$

Following from this definition is that:

$$\prec_{polgw} \stackrel{\text{def}}{\equiv} \prec_{polf-g} \wedge \approx_W \tag{4}$$

($Z \approx_W Z'$ is just alternative notation for $W = W'$.)
(3) and (4) imply that (Phase-g) is equivalent to:

$$MC(BPg; polgw; Z) \tag{5}$$

To see this, just compare the circumscription formulas for their elaborated definitions. Thus the phase policy is equivalent, in the context of the base, to a policy in which the function symbols W are explicitly fixed. Therefore, the functions W are fixed relative to the phase.

It thus remains to show only that the phase meets the other condition in quasi-propositionality: that the phase base implies domain closure for each of the phase's maximized defaults' default formulas. In the phase, there is only a single maximized default formula: Dg. The given quasi-propositionality of the global circumscription implies, by its definition, that the global base B implies domain closure on each of the global default formulas, and thus, in particular, domain closure on Dg. By (1), therefore, the phase base must also imply this domain closure on Dg.
QED Quasi-propositionality part of the Theorem

QED Theorem 5.35

E.5 For section 5.12: Safety of Higher-Priority Conclusions

Theorem 5.55 (Safety of Higher-Priority Conclusions)
In PDC and CLD: A default update is partially monotonic with respect to all conclusions derived solely from the previous higher-priority defaults. More precisely: none of the previous conclusions need to be retracted that were derived solely (Definition 5.50) from the previous defaults that are higher-priority than the new default according to the prioritization after the update.

More generally, a module update is partially monotonic with respect to all conclusions derived solely from the previous higher-priority modules (e.g., defaults).

Proof Overview: Non-trivial. Uses canonical conjunctive decomposition: Theorem 5.14. The essence is to use Observation 4.6: to consider the difference between the canonical decomposition before and after the update. All of the slices corresponding to the higher-priority defaults are left unchanged by the update. Why? The new update default(s) is irrelevant context (in the sense of Observations 4.8) relative to those slices, because it is lower-priority relative to them. There is a **subtlety**, however. **The global prioritization even among the previous defaults may change: by transitivity "through" the new default(s).** According to the prioritization information in the update, one of the previous defaults (say, j) may be higher than a new default (say, k), and another (say, l) lower than that same new default (k), without there previously having been any strict priority between the previous two. That is, j may have been incomparable to l previously, yet be strictly higher after the update. Nevertheless, by exploiting, in effect, the monotonicity of prioritization (Theorem 2.58), we are able to show that the slices corresponding to the higher-priority defaults are affected monotonically by the update. The details are a bit tricky; see Appendix for the (unfortunately somewhat tedious) details.

□

Proof of Theorem 5.55 :

Synopsis:

When we examine the slices corresponding to the higher-priority pre-orders, they are left unchanged because the prioritization on not just the dominators of the new module, but also on the incomparables of the new module, is left unchanged. Thus not just the fixing, but also the guarding, involved in the protection policy tactic $polf-i$ applied to create those slices, is untouched by the update. Proving this carefully takes some lemmas and hair.

Body of Proof:

Let u stand for the index of the new module pre-order. We write the previous theory as:

$$C0 \stackrel{\text{def}}{\equiv} C(B; H0^{N0}; R0; Z) \tag{D1}$$

and the new theory as:

$$C1 \stackrel{\text{def}}{\equiv} C(B; H1^{N1}; R1; Z) \tag{D2}$$

where

$$N1 \stackrel{\text{def}}{=} N0 \cup \{u\} \tag{D3}$$

$$H1^{N0} \stackrel{\text{def}}{=} H0^{N0} \tag{D4}$$

As we discussed when defining the concept of a module update (Definition 5.54), the new prioritization may be increased, but not decreased on the old index set, so that

$$R0^{N0} \leq R1^{N0} \tag{1}$$

Local notation: we write $Da(b)$ to stand for $Ra_D(b)$ and $Ia(b)$ to stand for $Ra_I(b)$, for $a = 0, 1$, and $b = i, u$.

Consider the dominator set of the new module's index, i.e., the set of indices of the previous higher-priority starting pre-orders:

$$S \stackrel{\text{def}}{=} D1(u) \tag{D5}$$

Note that

$$S \subseteq N0 \tag{2}$$

since S, by its definition, cannot include u.
Since every index in S is higher-priority, in $R1$, than u, it cannot be that $R1$ on S differs from $R0$ on S: the characteristic property of a module update implies that the only way that prioritization is increased in $R1^{N0}$ relative to $R0^{N0}$ is via transitivity "through" the new index. (Recall the discussion of transitivity "through" in the Proof Overview.) Thus

$$R1^S \equiv R0^S \tag{3.1}$$

Indeed, for the same reason, no pair of indices in $R_{DI}(u)$, which we will write as $DI1(u)$, can have their relative prioritization affected by the update:

$$R1^{DI1(u)} \equiv R0^{DI1(u)} \tag{3.2}$$

so that, also

$$\forall i \in DI1(u). \, (D1(i) = D0(i)) \;\wedge\; (I1(i) = I0(i)) \tag{3.3}$$

By Lemma E.9 (part (1) there),

$$S \text{ is up-closed} \tag{4}$$

relative to $R1$, and thus, by (3.1), relative to $R0$.

We are now ready to define the higher-priority previous conclusions. Let

$$C0\text{–}S \overset{\text{def}}{\equiv} C(B; H0^S; R0^S; \mathit{guard}\,(H0^{N0}; R0^{N0}); Z) \tag{D6}$$

$C0$–S is the upper-priority sub-theory corresponding to S in the previous theory. I.e., we applied $pol - S$ to the previous policy.
Similarly, we define

$$C1\text{–}S \overset{\text{def}}{\equiv} C(B; H1^S; R1^S; \mathit{guard}\,(H1^{N1}; R1^{N1}); Z) \tag{D7}$$

$C1$–S is the upper-priority sub-theory corresponding to S in the new theory. I.e., we applied $pol - S$ to the new policy.

We want to show

$$C1 \supset C0\text{–}S \tag{G1}$$

Assume

$$C1 \tag{A1}$$

then it suffices to show

$$C0\text{–}S \tag{G2}$$

By Theorem 5.14, (A1) $\iff$

$$\bigwedge_{i \in N1} C1\text{–}i \tag{5}$$

where $C1\text{–}i$ is the i^{th} slice of $C1$:

$$C1\text{–}i \;\stackrel{\text{def}}{\equiv}\; C(B; min\ H1_i; fix\ H1^{D1(i)}; guard\ (H1^{I1(i)}; R1^{I1(i)}); Z) \tag{D8}$$

(5), (2), (D3) $\Longrightarrow$

$$\bigwedge_{i \in S} C1\text{–}i \tag{6}$$

By Lemma E.25, (6) $\Longleftrightarrow$

$$C1\text{–}S \tag{7}$$

Consider the form of $C1\text{–}i$ for $i \in S$.
(D5), Lemma E.9 (parts (1) and (4) there) $\Longrightarrow$

$$\forall i \in S.\ DI1(i) \subseteq DI1(u) \tag{8}$$

(8), (3.2) $\Longrightarrow$

$$\forall i \in S.\ R1^{I1(i)} = R0^{I1(i)} \tag{9}$$

(D5) $\Longrightarrow$

$$\forall i \in S.\ u \in DB1(i) \tag{10}$$

(10) $\Longrightarrow$

$$\forall i \in S.\ (DI1(i) \subseteq N0) \ \wedge\ (i \in N0) \tag{11}$$

(11), (D4) $\Longrightarrow$

$$\forall i \in S.\ (H1^{DI1(i)} \ = \ H0^{DI1(i)}) \ \ \wedge \ \ (H1_i \ = \ H0_i) \tag{12}$$

(8), (3.3) $\Longrightarrow$

$$\forall i \in S.\ (D1(i) = D0(i)) \ \wedge\ (I1(i) = I0(i)) \tag{13}$$

(9), (12), (13) $\Longrightarrow$

$$\forall i \in S.\ C1\text{–}i \ \equiv\ C0\text{–}i \tag{14}$$

where $C0\text{–}i$ is the i^{th} slice of $C0$:

$$C0\text{–}i \;\stackrel{\text{def}}{\equiv}\; C(B; min\ H0_i; fix\ H0^{D0(i)}; guard\ (H0^{I0(i)}; R0^{I1(i)}); Z) \tag{D9}$$

(7), (14), Lemma E.25 $\Longrightarrow$ (G2).

QED Theorem 5.55

E.6 For section 5.13: Total Prioritization and Propositional Defaults

Lemma E.40 (Single Closed Default is Consistency)
In PDC without fixture (see discussion below): Maximizing a single closed default (or, likewise, minimizing a single propositional (0-ary) predicate) is equivalent to adding it (its negation) when it is consistent with the base. More precisely: We can formalize the consistency of $D[Z]$ with $B[Z]$ as:
$\exists Z'.\, B[Z'] \wedge D[Z']$
Suppose that $D[Z]$ is a single closed elementary formula, e.g., the negation of a single propositional predicate. Then

$$PDC(B; D; \emptyset; Z) \quad \equiv \quad B[Z] \;\wedge\; [(\exists Z'.\, B[Z'] \wedge D[Z']) \supset D[Z]]$$

(Equivalently, the rightmost $\supset$ above can be replaced by a $\equiv$.)

Note that the pre-conditions of this lemma *prohibit fixtures*, either explicit or implicit. This prohibition *includes function* symbols; this is an exception to our **assumption** (page 36) that all functions are fixed. Intuitively, fixture might block a default that is consistent from actually "going through".

Proof of Lemma E.40 : Synopsis: Short predicate calculus manipulation of the augmentation part of the circumscription. In detail:

$$PDC(B; D; \emptyset; Z) \tag{1}$$

$\overset{\text{def}}{\Longleftrightarrow}$

$$B[Z] \wedge \neg\exists Z'.\, B[Z'] \wedge D[Z] < D[Z'] \tag{2}$$

Because D is a single closed formula, (2) $\Longleftrightarrow$

$$B[Z] \wedge \neg\exists Z'.\, B[Z'] \wedge \neg D[Z] \wedge D[Z'] \tag{3}$$

$\Longleftrightarrow$

$$B[Z] \wedge \forall Z'.\, B[Z'] \wedge D[Z'] \;\supset\; D[Z] \tag{4}$$

$\Longleftrightarrow$

$$B[Z] \wedge (\exists Z'.\, B[Z'] \wedge D[Z']) \;\supset\; D[Z] \tag{5}$$

QED Lemma E.40

Notation: $C(B; H; Z)(Z')$**:**
For any Y that is similar to Z, we let $C(B; H; Z)(Y)$ stand for the result of substituting Y for Z in the circumscription formula $C(B; H; Z)$: i.e., for

$$B[Y] \wedge \neg\exists Z''.\, B[Z''] \wedge Z'' \prec_H Y$$

Definition E.41 (Unitary)

We say that a base sentence B is *unitary* with respect to pre-order H when

$$\forall Z, Z'.\, C(B;H;Z)(Z) \wedge C(B;H;Z)(Z') \quad \supset \quad Z \approx_H Z'$$

Discussion of Unitary:

Unitarity means that there is only one equivalence (with respect to H) class of maximally preferred models. Or, to make an analogy: in terms of Default Logic (see section 8.5), there is a unique "extension". More precisely: there is at most one equivalence class of maximally preferred models; the circumscription may not be satisfiable (recall our discussion of satisfiability in section 3.4). (In Default Logic, there also may not exist any "extension").

Lemma E.42

If B is unitary with respect to H, then

$$\forall Z, Z'.\, C(B;H;Z)(Z) \quad \supset \quad [C(B;H;Z)(Z') \; \equiv \; (B(Z') \wedge Z' \approx_H Z)] \tag{G}$$

Proof of Lemma E.42 : Elaborating the given unitarity condition:

$$\forall Z, Z'.\, C(B;H;Z)(Z) \wedge C(B;H;Z)(Z') \quad \supset \quad Z \approx_H Z' \tag{0}$$

we see that the left-to-right direction of the equivalence on the right-hand-side of (G) follows easily from (0) plus the fact that every circumscription implies its base. To show the right-to-left direction: Assume

$$C(B;H;Z)(Z) \tag{A1}$$

$$B(Z') \tag{A2}$$

$$Z' \approx_H Z \tag{A3}$$

We want to show

$$C(B;H;Z)(Z') \tag{G2}$$

(A2) $\implies$ it suffices to show

$$C_{aug}(B;H;Z)(Z') \tag{G3}$$

(A1) $\implies$ $C_{aug}(B;H;Z)(Z)$, i.e.:

$$\neg \exists Z''.\, B(Z'') \;\wedge\; Z'' \prec_H Z \tag{4}$$

We use an identity that follows from the transitivity of pre-orders:

$$\forall Z, Z', Z''.\, Z' \approx_H Z \quad \supset \quad [(Z'' \prec_H Z) \; \equiv \; (Z'' \prec_H Z')] \tag{5}$$

(A3), (5), (4) $\Longrightarrow$

$$\neg \exists Z''.\, B(Z'') \ \wedge\ Z'' \prec_H Z' \tag{G3}$$

QED Lemma E.42

Notation: $\gg$ for 2-ary strict prioritization:
We define $\gg$ as an infix notation for pre-orders. $\gg$ stands for "with strictly higher priority than". For any pre-orders H and L, $H \gg L$ is the pre-order that results from prioritizing H to be at strictly higher priority than L, where H and L are the starting pre-orders:

$$\preceq_{H \gg L} \ \stackrel{\text{def}}{\equiv}\ (\preceq_H \wedge (\approx_H \supset \preceq_L))$$

Lemma E.43 (Irrelevance of Protection when Unitary, Serial)

Let $\preceq_H$ and $\preceq_L$ be (arbitrary) pre-orders defined over Z.
Suppose the base sentence $B[Z]$ is unitary with respect to H. Then $C(B; H \gg L; Z)$ can be serially decomposed in descending sequence of prioritization **without** protecting the higher-priority pre-order H in the second phase:

$$\forall Z.\, C(B; H \gg L; Z) \ \equiv\ C(B; H; Z)/(L; Z) \tag{G}$$

Proof of Lemma E.43 : Canonical serial decomposition along prioritization in descending sequence, Theorem 5.32, implies that the left-hand-side of (G) $\iff$

$$C(B; H; Z) \ \wedge\ C(B; L; fix\ H; Z) \tag{1}$$

Using $C(B) \models B$, (1) $\iff$

$$C(B; H; Z) \ \wedge\ \neg \exists Z'.\, B(Z') \ \wedge\ Z' \approx_H Z \ \wedge\ Z' \prec_L Z \tag{2}$$

The right-hand-side of (G) $\stackrel{\text{def}}{\iff}$

$$C(B; H; Z)(Z) \ \wedge\ \neg \exists Z'.\, C(B; H; Z)(Z') \ \wedge\ Z' \prec_L Z \tag{3}$$

The equivalence of (2) and (3) follows from Lemma E.42.
QED Lemma E.43

Lemma E.44 (Maximizing a Single Propositional Default is Unitary)
Any base sentence is unitary with respect to a single-propositional-default pre-order.

Suppose $D[Z]$ is a single closed default (elementary) formula. Then any base sentence $B[Z]$ is unitary with respect to maximizing $\preceq_D$:

$$PDC(B; D; \emptyset; Z)(Z) \wedge PDC(B; D; \emptyset; Z)(Z'') \ \supset\ Z \approx_D Z'' \tag{G}$$

Note that there are **no fixtures** (permitted) in these circumscriptions: neither explicit nor implicit. In particular, **function symbols may not be fixed**; this is an exception to our **assumption** (page 36) that all functions are fixed. Intuitively, two maximally preferred models may differ on fixed symbols or formulas, even though they agree on whether or not D is satisfied.

Proof of Lemma E.44 :

Assume the two circumscriptions on the left-hand-side of (G). Then it suffices to show the right-hand-side of (G). By Lemma E.40, the first circumscription is equivalent to:

$$B[Z] \;\wedge\; [\neg D[Z] \supset (\neg \exists Z'.\, B[Z'] \wedge D[Z'])] \tag{1}$$

Likewise, the second circumscription is equivalent to:

$$B[Z''] \;\wedge\; [\neg D[Z''] \supset (\neg \exists Z'.\, B[Z'] \wedge D[Z'])] \tag{2}$$

Choosing Z' to be Z'' in (1), (1) implies

$$\neg D[Z] \;\supset\; \neg (B[Z''] \wedge D[Z'']) \tag{3}$$

(3) and the left-hand-side of (2) imply

$$\neg D[Z] \;\supset\; \neg D[Z''] \tag{4}$$

Likewise, choosing Z' to be Z in (2), (2) and the left-hand-side of (1) imply

$$\neg D[Z''] \;\supset\; \neg D[Z] \tag{5}$$

(4) and (5) $\quad \Longrightarrow$

$$D[Z] \equiv D[Z''] \tag{6}$$

(6) is just the right-hand-side of (G).

QED Lemma E.44

Theorem 5.69 (Clean Serial, Propositional Defaults)

In a propositional prioritized default circumscription $PDC(B; D; R; Z)$ without fixtures:
(By propositional, we mean that the base logical language is propositional.)
Suppose that the prioritization is a total order (cf. Definition 2.53).
Then the global PDC is serially decomposable into n *clean* phases, one per default, in each of which exactly that default is maximized. The sequencing is descending by priority. I.e., the first phase is the highest-priority default, and the last is the lowest-priority default.

$$PDC(B; D; R; Z) \;\equiv\; \underset{\substack{i \in N \\ via \\ R}}{///} \; PDC(\underset{B}{\bullet}\;; Di; \emptyset; Z)$$

where N is the index set of D. (Notationally, in the right-hand side PDC expression, we could have omitted the prioritization, which is empty, since there is only one default there.)

Moreover, each phase is just equivalent to a simple consistency check for that phase's default. That is, for each $i \in N$:

$$PDC(BBi; Di; \emptyset; Z) \;\equiv\; BBi[Z] \;\wedge\; \{(\exists Z'.\, BBi[Z'] \wedge Di[Z']) \;\supset\; D[Z]\}$$

where BBi is the base for the i^{th} phase. (For $i > 1$, BBi is the tier resulting from the $(i-1)^{th}$ phase.) Thus the global PDC can be computed with the following

Decidable Algorithm for Exhaustive Forward Inference:

0) Number the defaults' priority from 1 as highest to n as lowest. 1) Begin with $\mathcal{TIER}$ assigned to B. 2) For i from 1 to n: Test the consistency of Di against $\mathcal{TIER}$ using some satisfiability algorithm. (Note that propositional satisfiability is NP-hard.) If Di is consistent, revise $\mathcal{TIER}$ to include (i.e., conjoin) Di. (Then go on to the next i.) 3) The resulting $\mathcal{TIER}$ is the global circumscription. That is, it axiomatizes the circumscription, in the propositional base logical language $\mathcal{L}$: no second-order quantifiers involved. By saying that the algorithm performs "exhaustive" forward inference, we mean exhaustive in the non-monotonic aspect of reasoning. Only monotonic inference is required to use the result of the algorithm to arrive at any NM conclusion in the NM theory.

Generalization to Closed Defaults in First-Order Base Languages: This result generalizes to first-order (and, indeed, higher-order) base languages $\mathcal{L}$: as long as the defaults are all closed. That is, the default formulas must all be closed formulas.

The base sentence B is not required to be propositional; e.g., it may mention first-order (or even higher-order) quantifiers.

However, there is a further **subtlety: even functions may not be fixed**; this is an exception to our **assumption** (page 36) that all functions are fixed. This requirement is not terribly onerous. Including a complete theory of equality, e.g., domain closure plus the uniqueness of all names, in the base sentence suffices to effectively, though indirectly, fix the function symbols in a useful way, for example: see Theorem 2.24. Domain closure plus the uniqueness of all names suffices for the above result, for example, in place of no fixing of function symbols. However, in this case, note that the expressive power of the base language is effectively constrained to be no more than propositional. (By "effectively", we mean insofar as the circumscription is concerned.)

Of course for (full) first- and higher- order logic, consistency is not decidable. Thus the above algorithm becomes a procedure. The result of the procedure (if there is one) is an axiomatization of the circumscription, in the same logical language used to express the base sentence: e.g., an axiomatization in first-order if the base and default formulas were expressed in first-order.

Collapse to Propositional or First-Order: In short, in this case, circumscription **collapses** to the same expressive class as the base language $\mathcal{L}$ used to express the base and default formulas.. The circumscription is propositional if the base language is; it is first-order if the base language is.

Proof Overview: Non-trivial. Uses Corollary 5.61. Inductive, in the ascending direction of priority. Uses also three lemmas about how the propositional case behaves nicely. See Appendix.

$\square$

Proof of Theorem 5.69 :

Part I): Clean serial decomposition:

Without loss of generality, let:

$N \stackrel{\text{def}}{\equiv} \{1, \ldots, n\}$

$\forall i, j \in N.\ R(i, j) \stackrel{\text{def}}{\equiv} i < j$, where here (and only here) $<$ stands for ordinary arithmetic less-than.

Our proof of part I) is by induction in the ascending direction of priority. [2] Our inductive index is k, going from $k = 1$ to $k = n$. Let

$\forall k = 1, \ldots, n.\quad Nk \stackrel{\text{def}}{=} \{n - k + 1, \ldots, n\}$

so that

$N1 \stackrel{\text{def}}{=} \{n\}$

$Nn \stackrel{\text{def}}{=} N$

Our inductive hypothesis is that:

for any (elementary) base B,

$$PDC(B; D^{Nk}; R^{Nk}; Z) \equiv \underset{\substack{i \in Nk \\ via \\ R^{Nk}}}{///} PDC(\underset{B}{\bullet}; Di; \emptyset; Z) \qquad \text{(IHk)}$$

Initial case: inductive index $k = 1$.

$N1 = \{n\}$. The inductive hypothesis (IH1) is trivially true. **Inductive case:** Assume the inductive hypothesis for inductive index k: i.e., assume (IHk). Then we want to show the inductive hypothesis for inductive index $k + 1$, i.e., we want to show (IHk+1).

$$Nk + 1 \stackrel{\text{def}}{=} \{n - k\} \cup Nk$$

Thus

$$(D^{Nk+1}; R^{Nk+1}) \stackrel{\text{def}}{\equiv} (Dn - k; \emptyset) \gg (D^{Nk}; R^{Nk})$$

(Recall the $\gg$ notation introduced just before Lemma E.43.) That is, we can view the globally maximized pre-order for inductive index $k+1$ as the strict 2-ary prioritization of two modules: one at higher-priority containing the single default $Dn - k$, and a second at lower-priority containing the defaults D^{Nk}.

Because $Dn - k$ is a single-propositional-default pre-order, Lemmas E.43 and E.44 imply that this global (for inductive index $k + 1$) PDC can be serially decomposed along its prioritization, into one phase per module, where the second phase does **not protect** the higher-priority module's default:

[2] Note, therefore, that it is important that the number of defaults in the tuple D is finite; i.e., N must be finite.

$$PDC(B; D^{Nk+1}; R^{Nk+1}; Z) \;\equiv\; PDC(PDC(B; Dn-k; \emptyset; Z); D^{Nk}; R^{Nk}; Z) \tag{1}$$

(IHk) implies that the right-hand-side of (1) is equivalent to the right-hand-side of (IHk+1).
QED Part I)

Part II): Equivalence of each phase to a consistency check: This part of the Theorem follows directly from Lemma E.40. **QED Part II)**.

The algorithm / procedure in the Theorem statement follows immediately from parts I) and II).

QED Theorem 5.69

Lemma E.45 (Some Tautologies About Prioritized Default Pre-Orders)

Let $E[Z]$ be any tuple of open (or closed) (elementary) formulas, indexed by N. Let $TrueT$ stand for the tuple, similar to E, of identically true formulas. Let R be any prioritization partial order, defined over N.
Then the following are tautologies:

$$E[Z] \leq TrueT \tag{1}$$

$$E[Z] \preceq_R TrueT \tag{2}$$

$$\neg(E[Z] \succ_R TrueT) \tag{3}$$

$$(\neg(E[Z] \prec_R TrueT)) \;\equiv\; (E[Z] = TrueT) \tag{4}$$

Proof of Lemma E.45 :
(1) follows immediately from the definition of $\leq$.
(2) follows from (1) using the Lemma B.12.
(3) follows from (2) using the tautology $\preceq_R \;\supset\; \neg\succeq_R$.
The right-to-left direction of (4) follows from the reflexivity of pre-orders. The left-to-right direction of (4) follows from (3) plus Lemma B.10.
QED Lemma E.45

Lemma E.46 (Non-Conflicting Defaults Go Through)
In a prioritized default circumscription $PDC(B; D; R; Z)$ without fixture (see discussion below):
Suppose that every default in D is closed.
Suppose also that the defaults are collectively consistent with the base:

$$\exists Z.\, B[Z] \;\wedge\; \textstyle\bigwedge_{i \in N} Di[Z] \tag{0}$$

Intuitively, this condition says that the defaults do not conflict.
Then all the defaults "go through":

$$PDC(B;D;R;Z) \;\equiv\; B[Z] \;\wedge\; \bigwedge_{i \in N} Di[Z] \tag{G}$$

Observe that the prioritization R is thus irrelevant.

Note that the pre-conditions of this lemma *prohibit fixtures*, either explicit or implicit. This prohibition *includes function* symbols; this is an exception to our **assumption** (page 36) that all functions are fixed. Intuitively, fixture might block a default that is consistent from actually "going through". Also, remember by Lemma 3.42 that we can view any fixture as a pair of conflicting, polar defaults.

Proof of Lemma E.46 :

Left-to-right direction of (G):
Assume the left-hand-side of (G): this is defined as the conjunction of the following two sentences:

$$B[Z] \tag{1.1}$$

$$\neg\exists Z'. \;\; B[Z'] \;\wedge\; D[Z] \prec_R D[Z'] \tag{1.2}$$

Since any circumscription implies its base, it suffices to show :

$$\bigwedge_{i \in N} Di[Z] \tag{G2}$$

In (1.2), choose Z' to be Z^s, where Z^s is such that
$B[Z^s] \;\wedge\; \bigwedge_{i \in N} Di[Z^s]$
; (0) implies there must exist such a Z^s.
Thus (1.2) (with (0)) implies:

$$\neg(D[Z] \prec_R TrueT) \tag{2}$$

where $TrueT$ stands for the tuple, similar to D, of identically true formulas.
Lemma E.45 implies that (2) is equivalent to (G2).
QED left-to-right

Right-to-left direction of (G):
Assume the right-hand-side of (G):

$$B[Z] \tag{1.1}$$

$$\bigwedge_{i \in N} Di[Z] \tag{1.2}$$

Then it suffices to show the augmentation part of the circumscription on the right-hand-side of (G):

$$\neg\exists Z'. \;\; B[Z'] \;\wedge\; D[Z] \prec_R D[Z'] \tag{G2}$$

Lemma E.45 $\Longrightarrow$

$$\forall Z'. \, \neg(D[Z'] \succ_R TrueT) \tag{2}$$

(1.2) and (2) $\implies$

$$\forall Z'.\ \neg(D[Z'] \succ_R D[Z]) \tag{3}$$

(3) $\implies$ (G2). **QED Right-to-left**

QED Lemma E.46

Theorem 5.75 (For-Sure Beliefs As Highest-Priority Defaults)
In CLD without fixture axioms or fixed function symbols (see discussion of this subtlety below): When satisfiable, any group E of base axioms is equivalent to a highest-priority group of default axioms, one per base axiom in the group.

More precisely:
Let BB be the conjunction of the rest of the base axioms (if there are any such; if not, let $BB[Z]$ be $True$).
Suppose that E and BB are collectively satisfiable, or else that BB without E is unsatisfiable:

$$\exists Z.\ BB[Z] \ \models\ \exists Z.\ \ BB[Z] \wedge \textstyle\bigwedge_{i \in N} Ei[Z] \tag{0}$$

where N is the index set of the tuple E.
Then

$$PDC(BB \wedge (\textstyle\bigwedge_{i \in N} Ei[Z]); D; R; Z) \ \equiv\ PDC(BB; E, D; R2; Z) \tag{G}$$

Here, on the right-hand-side of (G): the defaults E form one module $(E; RI)$, the defaults D form another module $(D; R)$, and the global pre-order $(D, E; R2)$ is the result of prioritizing the E module at higher priority than the D module. Thus all of the defaults E have higher priority than all of the other defaults D. The prioritization RI internal to the higher-priority (E) module may be anything: it is irrelevant; e.g., it may be empty.

Note that the pre-conditions of this theorem *prohibit fixtures*. The reason is that fixture might block the defaults E from going through. This prohibition *includes function* symbols; this is an exception to our **assumption** (page 36) that all functions are fixed. This requirement is not terribly onerous: see Theorem 2.24. The **non-superfluity** of this prohibition on fixture is shown by the counterexamples in Observation 5.70. See also that Observation for more discussion of prohibition on fixture, including of functions.

Note that one can read this equivalence result in the other direction too: a collectively satisfiable, highest-priority group of closed defaults is just equivalent to their for-sure conjunction.

Proof Overview: The essence is: 1) Collective satisfiability means that the defaults E do not conflict with each other, and thus go through: we formalize this as a lemma. 2) Highest-priority means that they do not have to worry about conflicting with any of the other defaults. We combine serial decomposition for totally-prioritized groups, Corollary 5.61, with the lemma. See Appendix. □

Proof of Theorem 5.75 :

If BB is not satisfiable (i.e, if $\neg\exists Z.\ BB[Z]$), then both sides of (G) are false and the Theorem equivalence is trivially true. Thus let us assume BB is satisfiable:

$$\exists Z.\ BB[Z] \tag{A1}$$

(0) and (A1) imply that the sentences E are collectively satisfiable along with BB:

$$\exists Z.\quad BB[Z] \ \wedge\ \textstyle\bigwedge_{i \in N} Ei[Z] \tag{2}$$

Serial decomposition for totally-prioritized groups, Corollary 5.61, implies that the right-hand-side of (G) is equivalent to:

$$PDC(\ PDC(BB; E; RI; Z)\ ; D; R; fix\ E; Z) \tag{3}$$

(2) and Lemma E.46 imply that all of the defaults "go through" in the first phase: (3) is equivalent to

$$PDC(BB \wedge\ \textstyle\bigwedge_{i \in N} Ei\ ; D; R; fix\ E; Z) \tag{4}$$

Since all the sentences E are true in the base, fixing them makes no difference: Proposition 3.48 implies that (4) is equivalent to the left-hand-side of (G).

QED Theorem 5.75

Appendix F

Proofs for Chapter 6

F.1 For section 6.2: Disjoint Sub-Languages

Theorem 6.1 (Clean Decomposition, Disjoint Sub-Languages)
In a global $PDC(B; D^N; R; fix\ W; Z)$ where the only explicit fixing is of symbols (W): Let $Z \stackrel{\text{def}}{=} \langle Y, W \rangle$.
Suppose that $\{B1[Y1, W], \ldots, Bk[Yk, W]\}$ is a partition of the base axioms $B[Y, W]$, and that $\{D1[Y1, W], \ldots, Dk[Yk, W]\}$ is a partition of the default formulas $D[Y, W]$, where the predicate tuples $Y1, \ldots, Yk$ are a (disjoint) partition of Y. I.e., in terms of CLD, let there be a partition, of the base and default axioms, where the sub-languages used in each element of the partition are disjoint except possibly for fixed symbols. [1]
If the condition (0) (see below) on the prioritization is satisfied, then

$$PDC(B; D; R; fix\, W; Y, W) \equiv \bigwedge_{j=1}^{k} PDC(Bj; Dj; RIj; fix\, W; Y, W)$$

where Nj stands for the the index tuple, and $RIj \stackrel{\text{def}}{=} R^{Nj}$ stands for the internal prioritization, of the group of defaults Dj.
Note that the Y on the right-hand side above can, equivalently, be replaced by Yj, since, for each j:

$$PDC(B; Dj; RIj; fix\, W; Y, W) \equiv PDC(B; Dj; RIj; fix\, W; Yj, W)$$

The condition (0) on the prioritization R is defined as follows.
Either (0.1) R is the composition (Definition 2.51 and Theorem 2.56) of some external prioritization RE with the tuple RI of the internal prioritizations of each partition group of defaults (in this case we **define** R to be **modular** by partition) [2] ;

[1] If one relaxes the **assumption** (page 36) that all function symbols are fixed, then the overlap between partitions may include only the function (and predicate) symbols that are fixed. In that case, re-define Z,Y, and W to include function symbols as well as predicate symbols.

[2] Note that we are using partition in this section in a somewhat different sense than we did in section 5.13, where we spoke of partitioning the prioritization itself into 2 strictly-prioritized modules.

or, (0.2) R is layered;
or, more generally, (0.3) R has an "upper bound" RM that coincides with R on the partition groups and satisfies either (0.1) or (0.2).

Here, by bounding we mean that:

$R \leq RM$

and by coinciding we mean that:

$\forall j = 1, \ldots, k.\ R^{Nj} = RM^{Nj}$

Proof Overview: Surprisingly complicated. Uses canonical conjunctive decomposition (section 5.4): both hierarchical by modules, and one-by-one. Also uses the monotonicity of prioritization (Theorem 2.58).

The essence is to use the ability to separate existential quantifiers in the augmentation part of the circumscription. Non-layered prioritization makes this tricky: hence the prioritization conditions in the theorem. See Appendix. □

Proof of Theorem 6.1 : We first prove the two cases, (0.1) and (0.2), then the generalization to (0.3).

Local Notation: We find it convenient to use a slightly different notation in the proof than in the Theorem statement. Firstly, we leave the fixed symbols (W) implicit in the notation. Secondly, we prove the result for prioritized formula circumscription, in which elementary formulas are minimized, using the PFC notation. We thus write the global circumscription as:

$PFC(B; E; R; Y)$

Thirdly, we use Rj, instead of RIj to stand for the i^{th} member of the tuple of internal prioritizations RI. We thus write the j^{th} partition slice, on the right-hand-side of the Theorem, as:

$PFC(Bj; Ej; Rj; Y)$

(Equivalently, Y may be replaced by Yj.) Fourthly, we use RP, instead of RE, to stand for the prioritization between the partition modules. These changes avoid potentially-confusing overlap of symbols (e.g., D and I are used, in our usual abbreviated notation of prioritization partitions, to stand for Dominators and Incomparables), and help make things a bit briefer.

I) Given Layered Prioritization: case (0.2):

By our canonical conjunctive decomposition result, for the case of defaults one-by-one (Corollary 5.21) in guarded form:

$PFC(B; E; R; Y)$ is equivalent to

$$\bigwedge_{i \in N} PFC(B; E_i; \emptyset; fix\ E^{D(i)}; guard\ (E; R)^{I(i)}; Y) \tag{1}$$

where we write $D(i)$ for $R_D(i)$ and $I(i)$ for $R_I(i)$.

Consider the slice corresponding to index i. i is a member of exactly one partition: without loss of generality, we assume that $i \in N1$.

$\neg PFC_{aug}$ for slice i, the negated augmentation part of this slice circumscription, $\iff$

$$\exists Y1', Y2', \ldots, Yk'. \left(\bigwedge_{j=1}^{k} Bj(Yj')\right) \;\wedge\; E_i(Y1') < E_i(Y1) \\ \wedge\; E^{D(i)}(Y')=E^{D(i)}(Y) \\ \wedge\; E^{I(i)}(Y') \preceq_{R^{I(i)}} E^{I(i)}(Y) \tag{2}$$

Define

$$Dj(i) \;\stackrel{\text{def}}{\equiv}\; (D(i) \cap Nj) \qquad for\ j = 1, \ldots, k$$

$$Ij(i) \;\stackrel{\text{def}}{\equiv}\; (I(i) \cap Nj) \qquad for\ j = 1, \ldots, k$$

Then we can write

$$E^{D(i)}(Y')=E^{D(i)}(Y) \;\equiv \\ E^{D1(i)}(Y1')=E^{D1(i)}(Y1) \;\wedge\; \bigwedge_{j=2}^{k} E^{Dj(i)}(Yj')=E^{Dj(i)}(Yj) \tag{3}$$

By (0.2), R is layered. Thus

$$R^{I(i)} \;=\; \emptyset \tag{4}$$

$\implies$

$$R1^{I1(i)} \;=\; R^{I1(i)} \;=\; \emptyset \tag{5}$$

(5) $\implies$

$$E^{I(i)}(Y') \preceq_{R^{I(i)}} E^{I(i)}(Y) \;\equiv \\ E^{I1(i)}(Y1') \preceq_{R1^{I1(i)}} E^{I1(i)}(Y1) \;\wedge\; \bigwedge_{j=2}^{k} E^{Ij(i)}(Yj') \leq E^{Ij(i)}(Yj) \tag{6}$$

(3), (6) imply that (2) $\iff$

$$\exists Y1', Y2', \ldots, Yk'. \left(\bigwedge_{j=1}^{k} Bj(Yj')\right) \;\wedge\; POLF1i(Y1', Y1) \;\wedge\; GOi(Y', Y) \tag{7}$$

where

$$POLF1i(Y1', Y1) \;\stackrel{\text{def}}{\equiv}\; E_i(Y1') < E_i(Y1) \;\wedge\; E^{D1(i)}(Y1')=E^{D1(i)}(Y1) \\ \wedge\; E^{I1(i)}(Y1') \preceq_{R1^{I1(i)}} E^{I1(i)}(Y1)$$

is the policy relation (Definition E.4) for the slice, for index i, of the "partition 1 circumscription" $PFC(B; E1; R1; Y)$; and

$$GOi(Y', Y) \;\stackrel{\text{def}}{\equiv}\; \bigwedge_{j=2}^{k} E^{Dj(i)}(Yj')=E^{Dj(i)}(Yj) \;\;\wedge\;\; \bigwedge_{j=2}^{k} E^{Ij(i)}(Yj') \leq E^{Ij(i)}(Yj)$$

represents a Guard expression in the Other partitions' defaults. Since $GOi(Y', Y)$ does not mention $Y1'$ or $Y1$, (7) $\iff$

$$\exists Y1'.\, B1(Y1') \ \wedge\ POLF1i(Y1', Y1) \ \ \wedge\ \ Ai(Y) \tag{8}$$

where $Ai(Y)$ is defined as

$$\exists Y2', \ldots, Yk'.\ \left(\bigwedge_{j=2}^{k} Bj(Yj')\right) \ \wedge\ GOi(Y', Y) \tag{9}$$

Substituting $Y2$ for $Y2'$, ..., Yk for Yk' in (9) $\implies$

$$\left(\bigwedge_{j=2}^{k} Bj(Yj)\right) \ \supset\ \ Ai(Y) \tag{10}$$

Thus (10) and the equivalence of (2) to (8) $\implies$

$$\begin{aligned} B(Y) \ \supset & \\ & \{\, [PFC_{aug} \text{ for slice } i \text{ of } PFC(B;E;R;Y)] \ \equiv \\ & \qquad [PFC_{aug} \text{ for slice } i \text{ of } PFC(B1;E1;R1;Y)]\,\} \end{aligned} \tag{11}$$

$\implies$

$$\begin{aligned} B(Y) \ \supset & \\ & \{\, [\text{ slice } i \text{ of } PFC(B;E;R;Y)] \ \equiv \\ & \qquad [\text{ slice } i \text{ of } PFC(B1;E1;R1;Y)]\,\} \end{aligned} \tag{12}$$

Using (12) for each $i \in N1$, and using canonical conjunctive decomposition, one-by-one cf. Corollary 5.21, on the right-hand-side : $\implies$

$$B(Y) \ \supset\ \ \{\, [\textstyle\bigwedge_{i \in N1} \text{slice } i \text{ of } PFC(B;E;R;Y)] \ \equiv\ [PFC(B1;E1;R1;Y)]\,\} \tag{13}$$

Our choice of the partition index 1 as our focus above, was arbitrary. Thus (13) holds with j substituted for 1. Using it for each partition index $j = 1, \ldots, k$, with, again, canonical conjunctive decomposition cf. Corollary 5.21: $\implies$

$$B(Y) \ \supset\ \{\, PFC(B;E;R;Y) \ \equiv\ \bigwedge_{j=1}^{k} PFC(Bj;Ej;Rj;Y)\,\} \tag{14}$$

Using the fact that $C(B) \supset B$ for every circumscription,
plus the fact that $\left(\bigwedge_{j=1}^{k} Bj(Y)\right) \ \equiv\ B(Y)$: $\implies$

$$PFC(B;E;R;Y) \ \equiv\ \bigwedge_{j=1}^{k} PFC(Bj;Ej;Rj;Y) \tag{G}$$

as desired.

QED I)

II) Given that the Partitions Correspond to Modules: case (0.1):

Let j be a particular partition index. Since the partitions correspond to modules, i.e., by (0.1), by using the prioritization RP defined over the partition indices, we can divide up the partition indices

($\{1,\ldots,k\}$), according to their prioritization relative to j, into: $RP_D(j)$, $RP_I(j)$, and $RP_{DB}(j)$. We write these more simply as, respectively: $DD(j)$, $II(j)$, and $DBDB(j)$.

Consider the canonical conjunctive decomposition along prioritization of $PFC(B;E;R;Y)$, but this time hierarchical: using RP as the prioritization, rather than R. That is, consider a coarser-grain decomposition than in **I)**, by modules, into one slice per partition module: using Corollary 5.28. In particular, consider the slice of that decomposition corresponding to the partition index j. Without loss of generality, let us assume j is 1. $\neg PFC_{aug}$ for that slice is then:

$$\begin{aligned} \exists Y1', Y2', \ldots, Yk'.\ \left(\textstyle\bigwedge_{i=1}^{k} Bi(Yi')\right) \ \wedge\ & E1(Y1') \prec_{R1} E1(Y1) \\ \wedge\ & E^{DD(1)}(Y') = E^{DD(1)}(Y) \\ \wedge\ & E^{II(1)}(Y') \preceq_{R^{II(1)}} E^{II(1)}(Y) \end{aligned} \tag{1}$$

Since $DD(1)$ and $II(1)$ are disjoint from 1, (1) $\iff$

$$\exists Y1'.\ B1(Y1) \ \wedge\ E1(Y1') \prec_{R1} E1(Y1) \ \wedge\ AA1(Y) \tag{2}$$

where $AA1(Y)$ is defined as

$$\begin{aligned} \exists Y2', \ldots, Yk'.\ \left(\textstyle\bigwedge_{i=2}^{k} Bi(Yi')\right) \ \wedge\ & E^{DD(1)}(Y') = E^{DD(1)}(Y) \\ \wedge\ & E^{II(1)}(Y') \preceq_{R^{II(1)}} E^{II(1)}(Y) \end{aligned} \tag{3}$$

As in **I)**:

Substituting $Y2$ for $Y2'$, ..., Yk for Yk' in (3) $\implies$

$$\left(\textstyle\bigwedge_{i=2}^{k} Bi(Yi)\right) \ \supset\ AA1(Y) \tag{4}$$

Thus (4) and the equivalence of (1) to (2) $\implies$

$$\begin{aligned} & B(Y) \ \supset \\ & \quad \{\, [PFC_{aug} \text{ for } \textit{partition} \text{ slice 1 of } PFC(B;E;R;Y)] \ \equiv \\ & \qquad\qquad [PFC_{aug}(B1;E1;R1;Y)] \,\} \end{aligned} \tag{5}$$

Our choice of j as 1 was arbitrary. Thus (5) holds with j substituted for 1. Using it for each partition index $j \in \{1,\ldots,k\}$, plus canonical conjunctive decomposition by modules cf. Corollary 5.28 (along RP, not R), $\implies$

$$B(Y) \ \supset\ \{\, PFC(B;E;R;Y) \ \equiv\ \textstyle\bigwedge_{j=1}^{k} PFC(Bj;Ej;Rj;Y) \,\} \tag{6}$$

which implies (G), as in **I)**.

QED II)

III) Given that there is an "Upper Bound" satisfying (0.1) or (0.2): case (0.3):

For the proof of this case, we use (construct) a "lower bound" global prioritization, in addition to the given "upper bound". In the notation of the Theorem statement: let $RL \stackrel{\text{def}}{=} RE \circ RI$. (Recall $\circ$ notation for composition of partial orders, from Definition 2.51.) Then RL obeys condition (0.2), and

$$RL \leq R \leq RM$$

The monotonicity of prioritization (Theorem 2.58) implies that:

$$PFC(B;E;RM;Y) \implies PFC(B;E;R;Y) \implies PFC(B;E;RL;Y) \tag{1}$$

By **I)** and **II)**, the fact that RL and RM each obey (0.1) or (0.2) implies that we can apply the Theorem to them:

$$PFC(B;E;RL;Y) \equiv \bigwedge_{j=1}^{k} PFC(Bj;Ej;RLj;Y) \tag{2}$$

$$PFC(B;E;RM;Y) \equiv \bigwedge_{j=1}^{k} PFC(Bj;Ej;RMj;Y) \tag{3}$$

where RLj stands for RL^{Nj} and RMj stands for RM^{Nj}. But RL and RM coincide with R on the partitions, so that:

$$RLj = Rj = RMj \tag{4}$$

Thus the clean decompositions (one slice per partition) of the "upper bound" circumscription and of the "lower bound" circumscription are equivalent. "Sandwiched in-between", the original global circumscription must be equivalent to that clean decomposition.
(1), (2), (3), (4) $\implies$ (G).
QED III)
QED Theorem 6.1

Corollary 6.2 (Local Completeness of Inference, Disjoint Sub-Lang.)
In Theorem 6.1, suppose also that the fixed symbols too (including function symbols as well as predicate symbols) are partitionable in the same manner as the base and default axioms. Suppose furthermore that the the global circumscription is satisfiable. (If it is not, then inference is pretty pointless anyway.)
Then the j^{th} slice is sound and complete, relative to the global theory, for inference over its corresponding sub-language. That sub-language consists of the formulas that mention only the symbols $\langle Yj, Wj \rangle$. In other words, that slice entails such a formula if and only if it is entailed by the global theory. Note that this local soundness and completeness holds both for backward inference, e.g., query-answering, and for forward inference.

More generally, to perform inference using any subset Y of the symbols Z, one need only work in the conjunctive combination of those slices whose predicates cover that subset Y.

Proof Overview: Short. Uses existential projection applied to Theorem 6.1. See Appendix. $\Box$

Proof of Corollary 6.2 : It suffices to show that the projection of the whole global circumscription onto the sub-language is equivalent to the j^{th} slice.

Let $Z \stackrel{\text{def}}{=} \langle Y, W \rangle$ and $Zj \stackrel{\text{def}}{=} \langle Yj, Wj \rangle$. Without loss of generality, assume $j = 1$. Let Z^{-1} stand for $\langle Z2, \ldots, Zk \rangle$. We can express the projection via $\exists Z^{-1}$. . Consider

$$\exists Z^{-1}.\ PDC(B;D;R;Z) \tag{1}$$

By Theorem 6.1, $\iff$

$$\exists Z^{-1}.\ \bigwedge_{j=1}^{k} PDC(Bj; Dj; Rj; Zj)$$

$\iff$

$$PDC(B1; D1; R1; Z1) \ \wedge\ \exists Z^{-1}.\ \bigwedge_{j=2}^{k} PDC(Bj; Dj; Rj; Zj) \tag{2}$$

By our assumption of satisfiability of each slice, the right-hand-side of (2) is $True$. Thus (1) is equivalent to the left-hand-side of (2).

QED Corollary 6.2

F.2 For section 6.4: Conservative Extension

Theorem 6.8 (Conservative Updates: Clean Decomposition and Safety)
Let $PDC(B; D; R; fix\ F; Z)$ be a previous global PDC.
We consider updating in terms of CLD.

Suppose the (elementary) **sentence** $E[Y, Z]$ conservatively extends (see Definition 3.6) the previous base $B[Z]$. Here Y are new symbols: they are distinct from Z.

Then E is **globally monotonic as a base update, or as a default update with any priority**. Moreover, in the case where E is a default, the default "goes through".

In more detail: the circumscription after the base update is equivalent to: the circumscription before the update, conjoined with E:

$$PDC(B \wedge E; D; R; fix\ F; Y, Z) \ \equiv\ E[Y, Z] \wedge PDC(B; D; R; fix\ F; Z)$$

Likewise, the circumscription after the default update, in the case when the default update does *not* include prioritization, is the same as in the case of a base update:

$$PDC(B; D, E; R2; fix\ F; Y, Z) \ \equiv\ E[Y, Z] \wedge PDC(B; D; R; fix\ F; Z)$$

where $R2^N = R^N$ and N indexes D; i.e., where the new prioritization agrees with the old on the previous defaults, and is empty elsewhere.

In the case where the default update *does* include prioritization, the circumscription after the update is at least that strong:

$$PDC(B; D, E; R2; fix\ F; Y, Z) \ \supset\ E[Y, Z] \wedge PDC(B; D; R; fix\ F; Z)$$

More generally, we consider an update containing **several defaults**, and permit those defaults to be **open**. Let E^{NU} be a tuple of open (or closed) *formulas* Ei. The conjunctive collection of the universal closures of these formulas is defined as:

$$EC \ \stackrel{\text{def}}{\equiv}\ \bigwedge_{i \in NU} \forall Ei$$

Suppose that EC conservatively extends B:

$$B[Z] \ \models\ \exists Y.\ EC[Y, Z]$$

Then an update consisting of one default axiom per Ei, with any prioritization, is globally monotonic. Moreover, each of the defaults "goes through". In the case where the update does *not* include prioritization, the circumscription after is equivalent to the circumscription before conjoined with their (universal closures') conjunction.

$$PDC(B;D,E;R2;fix\ F;Y,Z) \equiv EC[Y,Z] \wedge PDC(B;D;R;fix\ F;Z)$$

where $R2^N = R^N$ and N indexes D; i.e., where the new prioritization agrees with the old on the previous defaults, and is empty elsewhere.

In the case where the default update *does* include prioritization, the circumscription after the update is at least that strong:

$$PDC(B;D,E;R2;fix\ F;Y,Z) \supset E[Y,Z] \wedge PDC(B;D;R;fix\ F;Z)$$

Above, the cases without prioritization updating can be viewed as a **clean conjunctive decomposition** in which there are two slices. The first slice's axiom set is the entire previous axiom set. The second slice's axiom set is the update. Likewise, the above can be viewed as a **clean serial decomposition**, with the same constituent axiom sets, in either sequence.

The non-conflict in this conservative extension case has mixed grain: it is between one or a few (the update) and a larger group (the previous axioms). Note that conservative extension is a restriction on the update, relative to (i.e., applicable to) *any* previous axiom set.

Proof Overview: For the default case, we use canonical conjunctive decomposition, one-by-one: Corollary 5.21. The case with prioritization follows from the case without prioritization: by the monotonicity of prioritization, Theorem 2.58. New prioritization axioms may affect the prioritization among the previous defaults: by transitivity "through" a new default (recall discussion of such effects in the proof of Theorem 5.55). See Appendix for details. □

Proof of Theorem 6.8 :

Base update:

$$PDC(B \wedge E;D;R;fix\ F;Z,Y) \tag{1}$$

$$\stackrel{\text{def}}{\Longleftrightarrow}$$

$$B[Z] \wedge E[Y,Z] \ \wedge\ \neg\exists Z',Y'.\ B[Z'] \wedge E[Y',Z'] \wedge Z \prec_{D;R} Z' \wedge Z \approx_F Z' \tag{2}$$

$$\Longleftrightarrow$$

$$B[Z] \wedge E[Y,Z] \ \wedge\ \neg\exists Z'.\ B[Z'] \wedge Z \prec_{D;R} Z' \wedge Z \approx_F Z' \wedge \exists Y'.\ E[Y',Z'] \tag{3}$$

The given conservative extension condition implies that (3) $\Longleftrightarrow$

$$B[Z] \wedge E[Y,Z] \ \wedge\ \neg\exists Z'.\ B[Z'] \wedge Z \prec_{D;R} Z' \wedge Z \approx_F Z' \tag{4}$$

(4) $\Longleftrightarrow$

$$E[Y,Z] \wedge PDC(B;D;R;fix\ F;Z) \tag{5}$$

Therefore, (1) $\Longleftrightarrow$ (5), as in the Theorem statement.

Clean Decomposition: The clean decompositions described in the Theorem statement follow immediately from the above.

QED Base update

Parallel Defaults update: We first show the case of updating with multiple open defaults but without new prioritization. The circumscription after updating is:

$$PDC(B; D, E; R2; fix\ F; Z, Y) \tag{1}$$

We apply canonical conjunctive decomposition for modules: Corollary 5.28. This implies that (1) is equivalent to the conjunction of two slices, one corresponding to the previous defaults, and a second corresponding to the new defaults. The first slice is:

$$PDC(B; D; R; fix\ F; guard\ (E; \emptyset); Z, Y) \tag{2.1}$$

The second slice is:

$$PDC(B; E; \emptyset; fix\ F; guard\ (D; R); Z, Y) \tag{2.2}$$

Definitionally elaborating, (2.1) plus (2.2) is equivalent to the conjunction of the following three sentences:

$$B[Z] \tag{3.1}$$

$$\neg \exists Z', Y'.\ B[Z'] \wedge Z \prec_{D;R} Z' \wedge Z \approx_F Z' \wedge E[Y, Z] \leq E[Y', Z'] \tag{3.2}$$

$$\neg \exists Z', Y'.\ B[Z'] \wedge Z \preceq_{D;R} Z' \wedge Z \approx_F Z' \wedge E[Y, Z] < E[Y', Z'] \tag{3.3}$$

(3.2) $\iff$

$$\neg \exists Z'.\ B[Z'] \wedge Z \prec_{D;R} Z' \wedge Z \approx_F Z' \wedge \exists Y'.\ E[Y, Z] \leq E[Y', Z'] \tag{4.2}$$

The given conservative extension condition is:

$$B[Z] \ \models\ \exists Y.\ EC[Y, Z] \tag{0}$$

But notice that:

$$\models\ EC[Y, Z] \ \equiv\ (E[Y, Z] = TrueT) \tag{5}$$

where $TrueT$ stands for a tuple, similar to the tuple E, each of whose elements is a formula that is identically true. I.e., the right-hand-side of (5) is:

$\forall i \in NU.\ \forall xi.\ Ei[Y, Z, xi] \equiv True$

(0) and (5) $\implies$

$$\models\ B[Z'] \ \supset\ \exists Y'.\ (E[Y', Z'] = TrueT) \tag{6}$$

(6) implies that (4.2) $\iff$

$$\neg \exists Z'.\ B[Z'] \wedge Z \prec_{D;R} Z' \tag{7.2}$$

But the conjunction of (3.1) and (7.2) is just the definitional elaboration of the circumscription before the update. Therefore, (3.1) plus (3.2) is equivalent to the circumscription before the update:

$PDC(B; D; R; fix\ F; Z)$

That is, in the first slice, the guarding of E makes no difference.

(3.3) $\iff$

$$\neg\exists Z'.\ B[Z'] \wedge Z \preceq_{D;R} Z' \wedge Z \approx_F Z' \wedge \exists Y'.\ E[Y, Z] < E[Y', Z'] \tag{8.3}$$

Choosing Z' to be Z in (8.3), (8.3) $\Longrightarrow$

$$\neg\{B[Z] \wedge \exists Y'.\ E[Y, Z] < E[Y', Z]\} \tag{9}$$

Here, we also used the reflexivity of pre-orders.

(3.1) and (9) $\Longrightarrow$

$$\neg\exists Y'.\ E[Y, Z] < E[Y', Z] \tag{10}$$

(0) and (5) $\Longrightarrow$

$$B[Z] \ \supset\ \exists Y'.\ (E[Y', Z] = TrueT) \tag{11}$$

(3.1), (9), and (11) $\Longrightarrow$

$$\neg(E[Y, Z] < TrueT) \tag{12}$$

$\iff$

$$(E[Y, Z] = TrueT) \tag{13}$$

(5) implies that (13) $\iff$

$$EC[Y, Z] \tag{14}$$

Thus (3.1) implies that (3.3) implies (14). But it is straightforward to check, using similar facts, that (3.1) also implies that (14) implies (3.3). Thus (3.1) implies that (3.3) is equivalent to (14). That is, in the presence of the second slice's base B, the augmentation part of the second slice is equivalent to EC.

Thus, the conjunction of the two slices is equivalent to:

$$PDC(B; D; R; fix\ F; Z)\ \wedge\ EC[Y, Z] \tag{15}$$

Therefore, (1) $\iff$ (15), as in the Theorem statement.

Clean Decomposition: Consider the clean decompositions described in the Theorem statement.

If B is not satisfiable, then the slice or phase containing the previous axioms is not satisfiable, and the overall circumscription after the update resulting from the conjunctive or serial combination is not satisfiable, either. Therefore, if B is not satisfiable, then the clean decompositions hold trivially.

Thus, let us assume that B is indeed satisfiable:

$$\exists Z.\ B[Z] \tag{A16}$$

The clean serial decomposition in which the update phase is second follows immediately from the above equivalence of (1) to (15).

For the clean conjunctive decomposition and the clean serial decomposition in which the update phase is first: we need to consider the circumscription formula corresponding to the CLD axiom set containing exactly the update. Local terminology: We call this "the update-only circumscription". It suffices to show the update-only circumscription is equivalent to EC:

$$PDC(True; E; \emptyset; Y, Z) \;\equiv\; EC[Y, Z] \tag{GCD}$$

(0) and (A16) $\implies$

$$\exists Y, Z.\ EC[Y, Z] \tag{17}$$

Elaborating its definition, the update-only circumscription $\stackrel{\text{def}}{\iff}$

$$\neg\exists Y', Z'.\ E[Y, Z] < E[Y', Z'] \tag{18}$$

Let $\langle Y^s, Z^s \rangle$ be such that $EC[Y^s, Z^s]$. (17) implies that there must be exist such a $\langle Y^s, Z^s \rangle$. In (18), choose Y' to be Y^s, and choose Z to be Z^s. Thus, (18) and (5) $\implies$

$$\neg(E[Y, Z] < TrueT) \tag{19}$$

(5) implies that (19) $\iff$

$$EC[Y, Z] \tag{20}$$

Thus, (18) implies (20). But it is easy to check that (20) also implies (18). Therefore, the update-only circumscription is equivalent to EC.

QED Parallel Defaults update

With New Prioritization: For the more general case where the update includes new prioritization as well as new defaults, use the monotonicity of prioritization, Theorem 2.58.

QED Theorem 6.8

F.3 For section 6.5: Sympathetically Solitary

Theorem 6.18 (Explicit Solution, Sympathetic-Solitary)
Suppose that a prioritized default circumscription is sympathetically solitary cf. Definition 6.17. Then

$$PDC(B; D; R; fix\ W; Q, Y, W) \;\equiv\; B[Q, Y, W] \;\wedge\; (D[Q, W] = E[W])$$

where E is the result of substituting $G[W]$ for σQ in D. (Equivalently, the $=$ above can be replaced by a $\geq$.)

Note that, as with the solitary and variable-solitary cases, the prioritization R is irrelevant.

Proof Overview: Uses analytic techniques and lemmas (about the properties of prioritized pre-orders) that we developed to prove our results about general-case canonical decomposition along prioritization. The argument involves substitution and properties of positivity / negativity. See Appendix. □

Proof of Theorem 6.18 :

We are given that:

$$D[Q, W] \text{ is positive in } \sigma Q \tag{0.1}$$

$$S[Q, Y, W] \text{ is positive in } \sigma Q \tag{0.2}$$

Since W is fixed, we can omit it from the second-order quantification in PDC_{aug}, the augmentation part of the PDC. Thus the PDC is equivalent to the conjunction of the following three sentences (1.1) – (1.3). (1.1) and (1.2) together constitute the base B. (1.3) is PDC_{aug}.

$$\sigma Q \leq G[W] \tag{1.1}$$

$$S[Q, Y, W] \tag{1.2}$$

$$\begin{aligned} \forall Y', Q'. \quad (\sigma Q' \leq G[W]) \ \wedge \ S[Q', Y', W] \ \wedge \ D[Q, W] {\preceq_R} D[Q', W] \\ \supset \ D[Q, W] {\approx_R} D[Q', W] \end{aligned} \tag{1.3}$$

Here, we used the following equivalent form of the augmentation part of a general circumscription:

$C_{aug}(B; H; Z) \ \equiv \ \forall Z'. \, B[Z'] \wedge Z' {\preceq_H} Z \ \supset \ Z' {\approx_H} Z$

First half: For the first half of the proof, we show the left-to-right direction of the equivalence in the Theorem statement. Assume the PDC, i.e., (1.1), (1.2), and (1.3). Since any circumscription implies its base, it suffices to show :

$$D[Q, W] = E[W] \tag{G1}$$

In (1.3), choose Q' to be $\sigma G[W]$, and choose Y' to be Y. Thus, (1.3) $\Longrightarrow$

$$S[\sigma G[W], Y, W] \ \wedge \ D[Q, W] {\preceq_R} E[W] \ \supset \ D[Q, W] {\approx_R} E[W] \tag{2}$$

Here, we used the definition of $E[W]$ (from the Theorem statement) and the fact that $sigma$ is its own inverse:

$\sigma(\sigma Q) \ = \ Q \quad$; and $\quad \sigma(\sigma G[W]) \ = \ G[W]$

(0.2), (1.1), and Fact B.11 $\Longrightarrow$

$$S[\sigma G[W], Y, W] \tag{3}$$

(0.1), (1.1), and Fact B.11 $\Longrightarrow$

$$D[Q, W] \leq E[W] \tag{4}$$

Lemma B.12 and (4) $\implies$

$$D[Q,W] \preceq_R E[W] \tag{5}$$

(2), (3), and (5) $\implies$

$$D[Q,W] \approx_R E[W] \tag{6}$$

Lemma B.10 and (6) $\implies$ (G1). **QED First half**

Second half: For the second half of the proof, we show the right-to-left direction of the equivalence in the Theorem statement. Assume $B \wedge (D = E)$, i.e., (1.1), (1.2), and (A7):

$$D[Q,W] = E[W] \tag{A7}$$

Then it suffices to show PDC_{aug}.
Recalling the elaboration of PDC_{aug} in (1.3):
(A7) and Lemma B.10 imply that $PDC_{aug} \iff$

$$\begin{aligned} \forall Y', Q'.\ \ (\sigma Q' \leq G[W]) \ \wedge \ S[Q',Y',W] \ \wedge \ E[W] \preceq_R D[Q',W] \\ \supset \ E[W] = D[Q',W] \end{aligned} \tag{G2}$$

Assume the left-hand-side of (G2):

$$\sigma Q' \leq G[W] \tag{A8}$$

$$S[Q',Y',W] \tag{A9}$$

$$E[W] \preceq_R D[Q',W] \tag{A10}$$

Then it suffices to show the right-hand-side of (G2):

$$E[W] = D[Q',W] \tag{G3}$$

By (0.1), $D[Q',W]$ is positive in $\sigma Q'$. Thus, (A8) and Fact B.11 $\implies$

$$D[Q',W] \leq E[W] \tag{11}$$

(11) and Lemma B.12 $\implies$

$$D[Q',W] \preceq_R E[W] \tag{12}$$

(A10) and (12) $\implies$

$$E[W] \approx_R D[Q',W] \tag{13}$$

(13) and Lemma B.10 $\Longrightarrow$ (G3). **QED Second half**

QED Theorem 6.18

Theorem 6.23 (Monotonicity of Sympathetic Default Update)
In Theorem 6.18, updating with a new default whose formula part is $U[Q,Y,W]$, with any prioritization for that new default relative to the previous defaults, is globally monotonic if U is sympathetic, i.e., positive in σQ (see property of S in Definition 6.17).

This result applies to abnormality theories as well, cf. Corollary 6.21.

Proof Overview: In addition to Theorem 6.18, we use canonical conjunctive decomposition for modules: Corollary 5.28. We first show the case where the new default is added in parallel to all the previous defaults, then use the monotonicity of prioritization: Theorem 2.58. See Appendix. □

Proof of Theorem 6.23 :

We describe updating in terms of CLD.

By the monotonicity of prioritization (Theorem 2.58), it suffices to show the case where the update includes no new prioritization, i.e., where the new default is in parallel with the previous defaults. Hence, we assume the update to be without prioritization. The circumscription after the update is:

$$PDC(B; D, U; R2; fix\ W; Q, Y, W) \tag{A}$$

where $R2^N = R^N$ and N indexes D; i.e., where the new prioritization agrees with the old on the previous defaults, and is empty elsewhere.

We want to show that the updated circumscription implies the previous circumscription. By Theorem 6.18, therefore, it suffices to show that (A) implies:

$$B[Q,Y,W] \ \wedge\ (D[Q,W] = E[W]) \tag{G1}$$

Assume (A). Since any circumscription implies its base, it suffices to show :

$$D[Q,W] = E[W] \tag{G2}$$

We use canonical conjunctive decomposition along prioritization, for modules: applied to the updated circumscription. We view the global (module) as the parallel prioritization of two (sub-)modules. The first module contains all of the previous defaults D. the second module contains only the new update default U. Corollary 5.28 thus implies that (A) is equivalent to the conjunction of two slices, one per module:

$$PDC(B; D; R; fix\ W; guard\ U; Q, Y, W) \tag{Slice1}$$

$$PDC(B; U; \emptyset; fix\ W; guard\ (D; R); Q, Y, W) \tag{Slice2}$$

We will show that the first slice (Slice1), corresponding to the previous defaults, implies (G2). The proof of this is quite similar to the first half of the proof of Theorem 6.18. We show that guarding U makes no difference, because of U's sympathy.

We are given that:

$$D[Q,W] \text{ is positive in } \sigma Q \tag{0.1}$$

$$S[Q,Y,W] \text{ is positive in } \sigma Q \tag{0.2}$$

$$U[Q,Y,W] \text{ is positive in } \sigma Q \tag{0.3}$$

Since W is fixed, we can omit it from the second-order quantification in PDC_{aug}, the augmentation part of the PDC that is (Slice1). Thus (Slice1) is equivalent to the conjunction of the following three sentences (1.1) – (1.3). (1.1) and (1.2) together constitute the base B. (1.3) is PDC_{aug} (for that slice).

$$\sigma Q \leq G[W] \tag{1.1}$$

$$S[Q,Y,W] \tag{1.2}$$

$$\begin{aligned}\forall Y', Q'.\ \ (\sigma Q' \leq G[W]) \ \wedge\ S[Q',Y',W] \ \wedge\ U[Q,Y,W] \leq U[Q',Y',W] \\ \wedge\ D[Q,W] \preceq_R D[Q',W] \\ \supset\ D[Q,W] \approx_R D[Q',W]\end{aligned} \tag{1.3}$$

Here, we used the following equivalent form of the augmentation part of a general guarded circumscription:

$$C_{aug}(B; H; guard\ G; Z) \ \equiv\ \forall Z'.\ B[Z'] \wedge Z' \preceq_G Z \wedge Z' \preceq_H Z \ \supset\ Z' \approx_H Z$$

In (1.3), choose Q' to be $\sigma G[W]$, and choose Y' to be Y. Thus, (1.3) $\implies$

$$\begin{aligned}S[\sigma G[W], Y, W] \ \wedge\ D[Q,W] \preceq_R E[W] \ \wedge\ U[Q,Y,W] \leq U[\sigma G[W], Y, W] \\ \supset\ D[Q,W] \approx_R E[W]\end{aligned} \tag{2}$$

Here, we used the definition of $E[W]$ (from the Theorem statement) and the fact that σ is its own inverse:

$$\sigma(\sigma Q) \ = \ Q \qquad ;\text{ and } \qquad \sigma(\sigma G[W]) \ = \ G[W]$$

(0.2), (1.1), and Fact B.11 $\implies$

$$S[\sigma G[W], Y, W] \tag{3}$$

(0.1), (1.1), and Fact B.11 $\implies$

$$D[Q,W] \leq E[W] \tag{4}$$

Lemma B.12 and (4) $\implies$

$$D[Q,W] \preceq_R E[W] \tag{5}$$

(0.3), (1.1), and Fact B.11 $\implies$

$$U[Q,Y,W] \leq U[\sigma G[W], Y, W] \tag{6}$$

(2), (3), (5), and (6) $\implies$

$$D[Q,W] \approx_R E[W] \tag{7}$$

Lemma B.10 and (7) $\implies$ (G2).

QED Theorem 6.23

F.4 For section 6.6: Strong Sympathy

Lemma F.1 (Formed Positively)
Let $D[Z]$ and $F[Z]$ are (finite) tuples of (elementary) formulas.
Suppose $E[Z]$ be formed purely and positively (Definition 3.50) from:
the formulas $D[Z]$; plus
formulas formed from $F[Z]$;
Then $E[Z]$ is positive with respect to the pre-order

$$\preceq_{D;\emptyset} \;\wedge\; \approx_F$$

Here, positivity with respect to a pre-order is in the sense of our generalization of positivity / negativity in Definition E.1.

Proof of Lemma F.1 : Let P and Q be tuples of (distinct) new symbols similar to D and F, respectively. Let us introduce axioms defining them as:

$$P = D[Z] \tag{Defn1}$$

$$Q = F[Z] \tag{Defn2}$$

Note that these explicit definitions are conservative in the sense of conservative extension: recall Definition 3.6. Fact 3.7 $\implies$

$$\models \;\; \forall Z.\;\; \exists P, Q.\;\; P = D[Z] \;\wedge\; Q = F[Z] \tag{CE}$$

Synopsis: The given forming condition about E implies that $E[Z]$ is equivalent to a formula $G[P,Q]$, purely in $\langle P,Q\rangle$, that has only positive appearances of P. G is positive with respect to the pre-order $\preceq_{P;\emptyset} \wedge \approx_Q$, which is just equivalent to $\preceq_{D;\emptyset} \wedge \approx_F$, therefore $E[Z]$ is.

Details:
We want to show that

$$\forall Z, Z'.\; Z \preceq_{D;\emptyset} Z' \wedge Z \approx_F Z' \;\;\supset\;\; Z \preceq_E Z' \tag{G1}$$

Local notation: We write D, F, D', F', E, G, E', and G' to stand for $D[Z]$, $F[Z]$, $D[Z']$, $F[Z']$, $E[Z]$, $G[P,Q]$, $E[Z']$, and $G[P',Q']$, respectively.
Using this notation, (G1) $\stackrel{\text{def}}{\iff}$

$$\forall Z, Z'.\; (D \leq D') \wedge (F{=}F') \;\;\supset\;\; (E \leq E') \tag{G2}$$

The given forming condition about E and Fact B.11 imply that

$$\forall P, Q, P', Q'.\; (P \leq P') \wedge (Q{=}Q') \;\;\supset\;\; (G \leq G') \tag{1}$$

(1) $\implies$

$$\forall Z, Z', P, Q, P', Q'.\ (P = D) \wedge (Q = F) \wedge (P' = D') \wedge (Q' = F') \supset\ [(P \leq P') \wedge (Q{=}Q')\ \supset\ (G \leq G')] \tag{2}$$

Next, we use a tautology:

$$\begin{aligned}\forall Z, Z', P, Q, P', Q'.\ (P = D) \wedge (Q = F) \wedge (P' = D') \wedge (Q' = F') \\ \supset\ \{[(P \leq P')\ \equiv\ (D \leq D')] \\ \wedge\ [(Q{=}Q')\ \equiv\ (F{=}F')] \\ \wedge\ [(G \leq G')\ \equiv\ (E \leq E')]\}\end{aligned} \tag{3}$$

(3) implies that (2) $\iff$

$$\forall Z, Z', P, Q, P', Q'.\ (P = D) \wedge (Q = F) \wedge (P' = D') \wedge (Q' = F') \supset\ [(D \leq D') \wedge (F{=}F')\ \supset\ (E \leq E')] \tag{4}$$

(4) $\iff$

$$\forall Z, Z'.\ [\exists P, Q, P', Q'.\ (P = D) \wedge (Q = F) \wedge (P' = D') \wedge (Q' = F')] \supset\ [(D \leq D') \wedge (F{=}F')\ \supset\ (E \leq E')] \tag{5}$$

(CE) implies that the left-hand-side of (5) is tautologically $True$.
Therefore: (5) $\iff$ (G2).

QED Lemma F.1

Theorem 6.29 (Monotonicity of Strongly Sympathetic Updates)
Let $PPC(B; P; \emptyset; \mathit{fix}\ W; P, Y, W)$ be a parallel predicate circumscription. Suppose that the sentence $U[P, W]$ is negative in the minimized predicates P, and mentions only P and the fixed symbols W (no auxiliaries may appear). Then updating the base with U is globally monotonic.

More generally, let $PDC(B; D; \emptyset; \mathit{fix}\ F; Z)$ be a parallel default circumscription. Suppose that the formula $U[Z]$ is tautologically equivalent to a formula that is formed purely and positively from the default formulas D plus fixed formulas. By "fixed formulas" here, we mean formulas that are fixed relative (Definition 3.47) to the circumscription: e.g., formulas formed from the explicitly fixed formulas F. (See also Definition 3.50.)

We **define** such a U to be **strongly sympathetic** to the circumscription, and to the defaults.

Then:

1) If U is closed, then updating the **base** with U is globally **monotonic**. In this case, the circumscription after the update is just U conjoined with the previous circumscription.

$$PDC(B \wedge U; D; \emptyset; \mathit{fix}\ F; Z)\ \equiv\ U[Z]\ \wedge\ PDC(B; D; \emptyset; \mathit{fix}\ F; Z)$$

And, interestingly:

2) Updating with U as a new **default** is **redundant** (Definition 4.17):

$$PDC(B; D, U; \emptyset; \mathit{fix}\ F; Z)\ \equiv\ PDC(B; D; \emptyset; \mathit{fix}\ F; Z)$$

Note that in the case of a default update, U may be an open formula.

Moreover, in terms of the special case of CLD without priorities, this monotonicity and redundancy are **forever** (Definition 4.17).

Note that the above depends on the **assumption** (page 36) that all functions are fixed. More generally, if only some functions are fixed, then, in the forming, only functions that are indeed fixed may appear. That is, the instantiation aspect of forming must be limited to employ only the subset of the functions that are indeed fixed.

For the case of a **base** update: the above can be viewed as a **clean conjunctive decomposition** in which there are two slices. In the first slice is the entire previous axiom set. In the second slice is the update. Likewise, the above can be viewed as a **clean serial decomposition**, with the same constituent axiom sets, in either sequence.

(For the case of a **default** update, the above can be viewed as a **clean serial decomposition**, with the update coming seqentially after the entire previous axiom set. This holds for any monotonic update.)

The non-conflict in this conservative extension case has mixed grain: it is between one (the update) and a group (the previous axioms). Note that strong sympathy is a restriction on the update, relative to (i.e., applicable to) *any* previous axiom set that is *parallel*.

Proof Overview: We use a generalization of positivity / negativity, and a lemma relating that to forming. In the default case, we show that the maximized global pre-order after the update is equivalent to that before the update. See Appendix. □

Proof of Theorem 6.29 :

Local notation: we omit prioritization since it is empty.

The key observation is that Lemma F.1, plus the given positive-forming condition, together imply that

$$\models \;\; Z \preceq_D Z \wedge Z \approx_F Z' \;\; \supset \;\; Z \preceq_U Z' \tag{KO}$$

Base update case:

The circumscription after updating the base with U is:

$$B[Z] \wedge U[Z] \;\wedge\; \neg\exists Z'.\, B[Z'] \wedge U[Z'] \wedge Z' \prec_D Z \wedge Z \approx_F Z' \tag{1}$$

$\iff$

$$B[Z] \wedge U[Z] \;\wedge\; \forall Z'.\, B[Z'] \wedge Z \preceq_D Z \wedge Z \approx_F Z' \;\; \supset \;\; (\neg U[Z'] \vee \neg(Z \approx_D Z')) \tag{2}$$

Since U is a closed formula, (KO) implies:

$$\forall Z, Z'.\, U[Z] \wedge Z \preceq_D Z' \wedge Z \approx_F Z' \;\; \supset \;\; U[Z'] \tag{3}$$

(3) implies that (2) is equivalent to:

$$B[Z] \wedge U[Z] \;\wedge\; \forall Z'.\, B[Z'] \wedge Z \preceq_D Z \wedge Z \approx_F Z' \;\; \supset \;\; \neg(Z \approx_D Z') \tag{4}$$

But (4) is equivalent to the conjunction of U with the circumscription before the update:

$$U[Z] \wedge PDC(B; D; \emptyset; fix\ F; Z) \tag{5}$$

Forever Monotonicity: Examining the above argument, we see that the global monotonicity (i.e., the equivalence of (1) to (5)) holds under any further CLD updates to the base and defaults and fixtures. Essentially, this is because (KO) will still hold later after such updating.
QED Base update case

Default update case:
Let DU stand for the tuple $\langle D, U\rangle$.
The circumscription after updating with the default U is:

$$B[Z] \wedge \neg\exists Z'.\, B[Z'] \wedge Z{\approx_F}Z' \wedge Z{\prec_{DU}}Z' \tag{1}$$

The circumscription before the update is:

$$B[Z] \wedge \neg\exists Z'.\, B[Z'] \wedge Z{\approx_F}Z' \wedge Z{\prec_D}Z' \tag{2}$$

We want to show that (1) is equivalent to (2).
(KO) implies

$$\forall Z, Z'.\, Z{\approx_F}Z' \;\supset\; (Z{\preceq_D}Z' \;\supset\; Z{\preceq_U}Z') \tag{3}$$

(3) $\implies$

$$\forall Z, Z'.\, Z{\approx_F}Z' \;\supset\; [(Z{\preceq_D}Z') \;\equiv\; (Z{\preceq_D}Z' \wedge Z{\preceq_U}Z')] \tag{4}$$

By symmetry (swap Z and Z'), (3) also $\implies$

$$\forall Z, Z'.\, Z{\approx_F}Z' \;\supset\; [(Z{\succeq_D}Z') \;\equiv\; (Z{\succeq_D}Z' \wedge Z{\succeq_U}Z')] \tag{5}$$

Note that

$$Z{\preceq_{DU}}Z' \;\stackrel{\text{def}}{\equiv}\; (Z{\preceq_D}Z' \wedge Z{\preceq_U}Z')$$

and also note that, for any pre-order H, the definition of $\prec_H$ is $\preceq_H \wedge \neg\succeq_H$.
Thus, (4) and (5) $\implies$

$$\forall Z, Z'.\, Z{\approx_F}Z' \;\supset\; (Z{\prec_D}Z' \;\equiv\; Z{\prec_{DU}}Z') \tag{6}$$

(6) implies that (1) is equivalent to (2).

Forever Redundancy: Examining the above argument, we see that the (global) redundancy (i.e., the equivalence of (1) to (2)) holds under any further CLD updates to the base and defaults and fixtures. Essentially, this is because (KO) will still hold later after such updating.
QED Default update case
QED Theorem 6.29

Bibliography

[Appelt and Konolige, 1988] Douglas Appelt and Kurt Konolige. A practical nonmonotonic theory for reasoning about speech acts. Technical Report Technical Note 432, SRI International, 333 Ravenswood Ave., Menlo Park, California 94025, Apr 1988.

[Apt *et al.*, 1987] K. Apt, H. Blair, and A. Walker. Towards a theory of declarative knowledge. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 89–148. Morgan Kaufmann, Los Altos, CA, 1987.

[Arrow, 1963] Kenneth J. Arrow. *Social Choice and Individual Values.* Yale University Press, 2nd edition, 1963.

[Bacchus, 1990] Fahiem Bacchus. *Representing and Reasoning with Probabilistic Knowledge.* MIT Press, Cambridge, Mass., 1990.

[Bacchus, 1991] Fahiem Bacchus. Default reasoning from statistics. *Proceedings of AAAI-91*, pages 392–398, 445 Burgess Drive, Menlo Park, CA 94025, 1991. AAAI Press.

[Bain and Muggleton, 1992] Michael Bain and Stephen Muggleton. Non-monotonic learning. In Stephen Muggleton, editor, *Inductive Logic Programming*, pages 145–162. Academic Press, New York, NY, 1992.

[Baker and Ginsberg, 1989] A. Baker and M. Ginsberg. A theorem prover for prioritized circumscription. *Proceedings IJCAI-89*, pages 463–467, Detroit, MI., 1989.

[Bobrow and Winograd, 1977] Daniel G. Bobrow and Terry Winograd. An overview of KRL, a knowledge representation language. *Cognitive Science*, 1(1):3 and ff., 1977.

[Boddy *et al.*, 1989] Mark Boddy, Robert P. Goldman, Keiji Kanazawa, and Lynn A. Stein. Investigations of model-preference defaults. Working paper, submitted to Computational Intelligence journal., 1989.

[Bossu and Siegel, 1985] G. Bossu and P. Siegel. Saturation, non-monotonic reasoning and the closed-world assumption. *Artificial Intelligence*, 25:13–63, 1985.

[Brewka, 1989a] G. Brewka. Preferred subtheories: An extended logical framework for default reasoning. *Proceedings IJCAI-89*, pages 1043–1049, Detroit, Michigan, 1989.

[Brewka, 1989b] Gerhard Brewka. *Nonmonotonic Reasoning: From Theoretical Foundations to Efficient Computation.* PhD thesis, University of Hamburg, Germany, 1989.

[Brown and Shoham, 1989] Allen L. Brown, Jr. and Yoav Shoham. New results on semantical nonmonotonic reasoning. In M. Reinfrank *et al*, editor, *Proceedings of the Second International Workshop on Non-Monotonic Reasoning*, Berlin, Germany, 1989. Springer Lecture Notes on Computer Science.

[Cadoli and Schaerf, 1992] Marco Cadoli and Marco Schaerf. Approximate inference in propositional default logic and circumscription. *Working Notes of the Fourth International Workshop on Non-Monotonic Reasoning*, pages 79–86, Dipartimento di Informatica e Sistemistica, Universita di Roma "La Sapienza", via Salaria 113, I-00198 Rome, ITALY, 1992. Available as a report from the authors. Held at the Hawk Inn, Plymouth, Vermont, May 28–31, 1992. Co-chaired by David Etherington and Henry Kautz of AT&T Bell Labs. Sponsored in part by AT&T and AAAI.

[Charniak *et al.*, 1987] Eugene Charniak, Christopher Riesbeck, Drew V. McDermott, and James R. Meehan. *Artificial Intelligence Programming, second edition.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1987.

[Clark, 1978] K. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 293–322. Plenum Press, New York, 1978.

[Clocksin and Mellish, 1981] W. F. Clocksin and C.S. Mellish. *Programming in Prolog.* Springer-Verlag, 1981.

[de Kleer and Konolige, 1989] Johann de Kleer and Kurt Konolige. Eliminating the fixed predicates from a circumscription. *Artificial Intelligence*, 39:391–398, 1989.

[de Kleer, 1986a] J. de Kleer. An assumption-based truth maintenance system. *Artificial Intelligence*, 28:127–162, 1986.

[de Kleer, 1986b] Johan de Kleer. Extending the ATMS. *Artificial Intelligence*, 28:163–196, 1986.

[Delgrande, 1987a] J. Delgrande. An approach to default reasoning based on a first-order conditional logic. *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*, pages 340–345, Seattle, 1987.

[Delgrande, 1987b] James P. Delgrande. A first-order conditional logic for prototypical properties. *Artificial Intelligence*, 33:105–130, 1987.

[Delgrande, 1991] James P. Delgrande. Incorporating nonmonotonic reasoning in Horn clause theories. *Proceedings of AAAI-91*, pages 405–411, 445 Burgess Drive, Menlo Park, CA 94025, 1991. AAAI Press.

[Dietterich, 1982] Thomas G. Dietterich. Learning and inductive inference. In Paul R. Cohen and Edward A. Feigenbaum, editors, *The Handbook of Artificial Intelligence, Volume 3*, pages 323–512. Morgan Kaufmann, San Mateo, California, 1982.

[Dowling and Gallier, 1984] W.F. Dowling and J.H. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *Journal of Logic Programming*, 3:267–284, 1984.

[Doyle and Wellman, 1991] Jon Doyle and Michael P. Wellman. Impediments to universal preference-based default theories. *Artificial Intelligence*, 49:97–128, 1991. Revised and expanded from version in *Proceedings of the First International Conference on Principles of Knowledge Representation*, pp. 94-102, Toronto, Ontario, Morgan Kaufmann, 1989.

[Doyle, 1979] J. Doyle. A truth maintenance system. *Artificial Intelligence*, 12:231–272, 1979.

[Doyle, 1985] J. Doyle. Expert systems and the "myth" of symbolic reasoning. *IEEE Transactions on Software Engineering*, 11:1386–1390, 1985.

[Eiter and Gottlob, 1991] Tomas Eiter and Georg Gottlob. Propositional circumscription and extended closed world reasoning are π_2^P-complete. Technical report, Technical University of Vienna, Institute for Information Systems, Christian Dppler Laboratory for Expert Systems, Paniglgasse 16, A-1040 Wien, Austria, May 1991. Revised Technical Report CD-TR 91/20. To appear in *Theoretical Computer Science*.

[Enderton, 1972] Herbert B. Enderton. *A Mathematical Introduction to Logic.* Academic Press, New York, NY, 1972.

[Etherington and Reiter, 1983] D. Etherington and R. Reiter. On inheritance hierarchies with exceptions. *Proceedings of the Second National Conference on Artificial Intelligence (AAAI-83)*, pages 104–108, Washington, D.C., 1983.

[Etherington *et al.*, 1985] D. Etherington, R. Mercer, and R. Reiter. On the adequacy of predicate circumscription for closed-world reasoning. *Computational Intelligence*, 1:11–15, 1985.

[Etherington, 1987] David W. Etherington. Relating default logic and circumscription. *IJCAI-87.* Morgan Kaufmann, 1987.

[Etherington, 1988] D. Etherington. *Reasoning with Incomplete Information.* Pitman, London, 1988.

[Garey and Johnson, 1979] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, San Francisco, California, 1979.

[Geffner, 1992] Hector Geffner. *Default Reasoning: Causal and Conditional Theories.* MIT Press, Cambridge, Mass., 1992.

[Gelfond *et al.*, 1989] Michael Gelfond, Halina Przymusinska, and Teodor Przmusinski. On the relationship between circumscription and negation as failure. *Artificial Intelligence*, 38(1):75–94, 1989.

[Genesereth and Nilsson, 1987] M. Genesereth and N. Nilsson. *Logical Foundations of Artificial Intelligence.* Morgan Kaufmann, Los Altos, CA., 1987.

[Ginsberg, 1986] Matthew L. Ginsberg. Counterfactuals. *Artificial Intelligence*, 30:35–80, 1986.

[Ginsberg, 1987] M. Ginsberg, editor. *Readings in Nonmonotonic Reasoning.* Morgan Kaufmann, Los Altos, CA., 1987.

[Ginsberg, 1988] M. Ginsberg. Multivalued logics: A uniform approach to reasoning in artificial intelligence. *Computational Intelligence*, 4:265–316, 1988.

[Ginsberg, 1989] M. Ginsberg. A circumscriptive theorem prover. *Artificial Intelligence*, 39:209–230, 1989.

[Gottlob, 1992] Georg Gottlob. Complexity results for nonmonotonic logics. *Working Notes of the Fourth International Workshop on Non-Monotonic Reasoning*, pages 111–125, Technical University of Vienna, Institute for Information Systems, Christian Dppler Laboratory for Expert Systems, Paniglgasse 16, A-1040 Wien, Austria, 1992. Available as a report from the author. Shortened version of a paper with the same title to appear in *The Journal of Logic and Computation.* The Workshop was held at the Hawk Inn, Plymouth, Vermont, May 28–31, 1992. Co-chaired by David Etherington and Henry Kautz of AT&T Bell Labs. Sponsored in part by AT&T and AAAI.

[Grosof and Russell, 1990] Benjamin N. Grosof and Stuart J. Russell. Shift of bias as non-monotonic reasoning. In Pavel Brazdil and Kurt Konolige, editors, *Machine Learning, Meta-Reasoning, and Logics.* Kluwer Academic, 1990. Based on the workshop held in Sesimbra, Portugal, in Feb. 1988. Also available as IBM Research Report RC14620.

[Grosof, 1984] Benjamin N. Grosof. Default reasoning as circumscription. *Proceedings of the First AAAI Non-Monotonic Reasoning Workshop*, pages 115–124, Oct 1984. Held New Paltz, NY.

[Grosof, 1988] Benjamin N. Grosof. Non-monotonicity in probabilistic reasoning. In J. Lemmer and L. Kanal, editors, *Uncertainty in Artificial Intelligence 2*, pages 237–249. Elsevier Science Publishers, 1988. Volume containing revised versions of papers appearing in Proceedings of the Second International Workshop on Uncertainty in Artificial Intelligence, held Philadelphia, PA, August 1986.

[Grosof, 1991] Benjamin N. Grosof. Generalizing prioritization. *Proceedings of the Second International Conference on Principle of Knowledge Representation and Reasoning*, pages 289–300, April 1991. Also available as IBM Research Report RC15605, IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598.

[Grosof, 1992a] Benjamin N. Grosof. Defeasible and pointwise prioritization (working title). Working paper., April 1992.

[Grosof, 1992b] Benjamin N. Grosof. Generalizing prioritization II (working title). Working paper., 1992.

[Grosof, 1992c] Benjamin N. Grosof. Reformulating non-monotonic theories for inference and updating. In Michael Lowry, editor, *Proceedings of the 1992 Workshop on Change of Representation and Problem Reformulation*, Moffett Field, California, April 1992. NASA Ames Research Center Technical Report FIA-92-06. Workshop held Asilomar, California, April 28 – May 1, 1992. Also available as IBM Research Report RC17955.

[Helft *et al.*, 1991] Nicolas Helft, Katsumi Inoue, and David Poole. Query answering in circumscription. *Proceedings of IJCAI-91*, pages 426–431, San Mateo, California, 1991. Morgan Kaufmann.

[Helft, 1989] Nicolas Helft. Induction as nonmonotonic inference. In Ronald J. Brachman, Hector J. Levesque, and Raymond Reiter, editors, *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, San Mateo, California, 1989. Morgan Kaufmann. Held Toronto, Ontario, Canada, May 1989.

[Imielinski, 1987] T. Imielinski. Results on translating defaults to circumscription. *Artificial Intelligence*, 32:131–146, 1987.

[Inoue and Helft, 1990] Katsumi Inoue and Nicolas Helft. On theorem provers for circumscription. *Proceedings of the Canadian Conference on Computer Science and Artificial Intelligence '90*, 1990. Held Ottawa, Canada, May 1990.

[Junker, 1991] Ulrich Junker. Prioritized defaults: Implementation by tms and applications to diagnosis. *Proceedings of IJCAI-91*, pages 310–315, San Mateo, California, 1991. Morgan Kaufmann. Held Sydney, Australia, August 1991.

[Kautz and Selman, 1989] H. Kautz and B. Selman. Hard problems for simple default logics. *Proceedings of the First International Conference on Principle of Knowledge Representation and Reasoning*, pages 189–197, Toronto, Ontario, 1989.

[Kautz and Selman, 1991] Henry Kautz and Bart Selman. A general framework for knowledge compilation. *Proceedings of the First International Workshop on Processing Declarative Knowledge*, AT&T Bell Labs, Murray Hill, NJ 07974, 1991. Available as a report from the authors. Held Kaiserslautern, Germany, July 1991.

[Kautz, 1986] Henry A. Kautz. The logic of persistence. *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86)*, pages 401–405. Morgan Kaufmann, 1986. Held in Philadelphia, PA.

[Kohavi and Shoham, 1992] Ron Kohavi and Yoav Shoham. Applications of Datalog theories in AI. Computer Science Dept., Stanford University, Stanford, California 94305, July 1992.

[Kolaitis and Papadimitriou, 1988] Phokion G. Kolaitis and Christos H. Papadimitriou. Some computational aspects of circumscription. *Proceedings of AAAI-88*, pages 465–469, San Mateo, California, 1988. Morgan Kaufmann. Held Minneapolis, MN.

[Konolige, 1988a] K. Konolige. On the relation between default logic and autoepistemic logic. *Artificial Intelligence*, 35:343–382, 1988.

[Konolige, 1988b] Kurt Konolige. Hierarchic autoepistemic theories for nonmonotonic reasoning. *Proceedings of AAAI-88*, pages 439–443. Morgan Kaufmann, 1988. Held Minneapolis, MN.

[Kowalski, 1979] Robert Kowalski. *Logic for Problem Solving.* North-Holland, 1979.

[Kraus *et al.*, 1990] S. Kraus, D. Lehmann, and M. Magidor. Preferential models and cumulative logics. *Artificial Intelligence*, 44:167–207, 1990.

[Lehmann and Magidor, 1988] D. Lehmann and M. Magidor. Rational logics and their models: a study in cumulative logic. Technical report, Dept. of Computer Science, Hebrew University, Jerusalem 91904, Israel, November 1988.

[Lehmann, 1989] D. Lehmann. What does a conditional knowledge base entail? *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pages 212–222, Toronto, Ontario, 1989. Morgan Kaufmann.

[Levy, 1992] Alon Levy. Irrelevance in problem-solving. Technical report, Stanford University Computer Science Department, Knowledge Systems Laboratory, Stanford, California 94305, 1992.

[Lifschitz, 1984] Vladimir Lifschitz. Some results on circumscription. *Proceedings of the First AAAI Non-Monotonic Reasoning Workshop*, pages 151–164, Oct 1984. Held New Paltz, NY.

[Lifschitz, 1985] V. Lifschitz. Computing circumscription. *Proceedings IJCAI-85*, pages 121–127, Los Angeles, CA, 1985.

[Lifschitz, 1986] V. Lifschitz. On the satisfiability of circumscription. *Artificial Intelligence*, 28:17–27, 1986.

[Lifschitz, 1987a] Vladimir Lifschitz. On the declarative semantics of logic programs with negation. In Matthew L. Ginsberg, editor, *Readings in Nonmonotonic Reasoning.* Morgan Kaufmann, San Mateo, CA, 1987.

[Lifschitz, 1987b] Vladimir Lifschitz. Pointwise circumscription. In Matthew L. Ginsberg, editor, *Readings in Nonmonotonic Reasoning.* Morgan Kaufmann, San Mateo, CA, 1987.

[Lifschitz, 1988a] V. Lifschitz. Circumscriptive theories: a logic-based framework for knowledge representation. *Journal of Philosophical Logic*, 17:391–441, 1988.

[Lifschitz, 1988b] V. Lifschitz. On the declarative semantics of logic programs. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 177–192. Morgan Kaufmann, Los Altos, CA., 1988.

[Lifschitz, 1989] Vladimir Lifschitz. Between circumscription and autoepistemic logic. *Proceedings of the First International Conference on Principle of Knowledge Representation and Reasoning*, pages 235–244, Toronto, Ontario, 1989.

[Lifschitz, 1990a] Vladimir Lifschitz. Circumscription. *Handbook of Logic in Artificial Intelligence and Logic Programming*. Oxford University Press, 1990.

[Lifschitz, 1990b] Vladimir Lifschitz. On open defaults. *Proceedings Symposium on Computational Logic*, Brussels, Belgium, 1990.

[Loui, 1987] R. Loui. Defeat among arguments: A system of defeasible inference. *Computational Intelligence*, 3(3):100–107, 1987.

[Loui, 1989] Ronald P. Loui. Analogical reasoning, defeasible reasoning, and the reference class. *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning (KR '89)*, pages 256–265, San Mateo, California, 1989. Morgan Kaufmann.

[Makinson, 1989] D. Makinson. General theory of cumulative inference. In M. Reinfrank *et al.*, editor, *Proceedings of the Second International Workshop on Non-Monotonic Reasoning*, pages 1–18, Berlin, Germany, 1989. Springer Lecture Notes on Computer Science.

[McCarthy and Hayes, 1969] J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Mitchie, editors, *Machine Intelligence 4*, pages 463–502. American Elsevier, New York, 1969.

[McCarthy, 1968] John McCarthy. Programs with common sense. In Marvin Minsky, editor, *Semantic Information Processing*, pages 403–409, Cambridge, Mass., 1968. MIT Press. Reprinted from original 1958 appearance in: Proceedings of the Symposium on the Mechanization of Thought Processes, National Physical Laboratory I:77–84.

[Mccarthy, 1977] John Mccarthy. Epistemological problems of artificial intelligence. *Proceedings of IJCAI-77*, pages 1038–1044, 1977. Held Cambridge, Mass.

[McCarthy, 1980] J. McCarthy. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

[McCarthy, 1986] J. McCarthy. Applications of circumscription to formalizing commonsense knowledge. *Artificial Intelligence*, 28:89–116, 1986.

[McCarthy, 1987] J. McCarthy. Generality in artificial intelligence. *Communications of the ACM*, 30, 1987.

[McDermott and Doyle, 1980] D. McDermott and J. Doyle. Non-monotonic logic I. *Artificial Intelligence*, 13:41–72, 1980.

[McRoy, 1989] Susan Weber McRoy. Nonmonotonic reasoning in natural language. Dept. of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 1A4., Nov 1989.

[Minksy, 1975] Marvin Minksy. A framework for representing knowledge. In P. Winston, editor, *The Psychology of Computer Vision*. McGraw Hill, 1975.

[Moore, 1985] R. Moore. Semantical considerations on non-monotonic logics. *Artificial Intelligence*, 25:75–94, 1985.

[Morgenstern and Guerriro, 1991] Leora Morgenstern and Ramiro Guerriro. Epistemic logics for multiple agent nonmonotonic reasoning I. Submitted. IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598. Includes revised version of contents of paper appearing in Proceedings of AAAI-90, entitled "A Formal Theory of Multiple Agent Nonmonotonic Reasoning", by Leora Morgenstern., Sep 1991.

[Muggleton, 1992] Stephen Muggleton, editor. *Inductive Logic Programming*. Academic Press, New York, NY, 1992.

[Nebel, 1989] Bernhard Nebel. A knowledge-level analysis of belief revision. *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pages 301–311, San Mateo, CA, 1989. Morgan Kaufmann. Held Toronto, Canada.

[Nilsson, 1983] Nils J. Nilsson. AI prepares for 2001. *AI Magazine*, 4(4), 1983. Based on AAAI-83 Presidential Address.

[Perrault, 1987] C. Raymond Perrault. An application of Default Logic to speech act theory. Technical Report CSLI-87-90, Center for the Study of Language and Information, Stanford University, Ventura Hall, Stanford University, Stanford, California 94305, Mar 1987.

[Pollock, 1987] J. Pollock. Defeasible reasoning. *Cognitive Science*, 11:481–518, 1987.

[Poole, 1988] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36:27–47, 1988.

[Przymusinski, 1988] Teodor Przymusinski. On the declarative semantics of deductive databases and logic programs. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*. Morgan Kaufmann, San Mateo, CA., 1988.

[Przymusinski, 1989] T. Przymusinski. An algorithm for circumscription. *Artificial Intelligence*, 38:49–73, 1989.

[Qian and Irani, 1991] Zhaogang Qian and Keki B. Irani. Circumscribing defaults. *Proceedings of IJCAI-91*, San Mateo, California, 1991. Morgan Kaufmann. Held Sydney, Australia, August 1991.

[Quine, 1969] Willard Van Orman Quine. *Set Theory And Its Logic, revised edition.* Harvard University Press, Cambridge, Mass., 1969.

[Rabinov, 1989] Arkady Rabinov. A generalization of collapsible cases of circumscription. *Artificial Intelligence*, 38(1):111–117, 1989.

[Raiffa, 1968] Howard Raiffa. *Decision Analysis: Introductory Lectures on Choices under Uncertainty.* Addison-Wesley, Reading, Mass., 1968.

[Raiman and de Kleer, 1992] Olivier Raiman and Johan de Kleer. A minimality maintenance system. *Working Notes of the 4th International Workshop on Non-Monotonic Reasoning*, 1992. Held Plymouth, Vermont, May 28–31, 1992. Sponsored by AAAI.

[Rathmann, 1989] Peter K. Rathmann. Circumscribing equality. *Proceedings of IJCAI-89*, pages 468–473, San Mateo, California, 1989. Morgan Kaufmann.

[Rathmann, 1990] Peter K. Rathmann. *Nonmonotonic Semantics for Partitioned Knowledge Bases.* PhD thesis, Computer Science Dept., Stanford University, Stanford, California 94305, June 1990.

[Reiter and Criscuolo, 1981] Raymond Reiter and Giovanni Criscuolo. On interacting defaults. *Proceedings of IJCAI-81*, pages 270–276, San Mateo, California, 1981. Morgan Kaufmann.

[Reiter and de Kleer, 1987] R. Reiter and J. de Kleer. Foundations of assumption-based truth maintenance systems: a preliminary report. *Proceedings AAAI-87*, pages 183–188, San Mateo, CA, 1987. Morgan Kaufmann. Held in Seattle, WA.

[Reiter, 1978] R. Reiter. On closed-world data bases. In H. Gallaire and J. Minker, editors, *Logic and data bases*, pages 55–76. Plenum Press, 1978.

[Reiter, 1980] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 12:81–132, 1980.

[Reiter, 1982] Raymond Reiter. Circumscription implies predicate completion (sometimes). *Proceedings of AAAI-82*, pages 418–420. American Association for Artificial Intelligence, Menlo Park, CA, 1982. Held Pittsburgh, PA.

[Reiter, 1987a] R. Reiter. Nonmonotonic reasoning. *Annual Review of Computer Science*, 2:147–186, 1987.

[Reiter, 1987b] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

[Russell and Grosof, 1987] Stuart J. Russell and Benjamin N. Grosof. A declarative approach to bias in concept learning. *Proceedings of AAAI-87*, pages 505–510, San Mateo, CA, 1987. Morgan Kaufmann. Held Seattle, WA. The first and second halves are complementary but distinct papers. The first is a preliminary version of [Russell and Grosof, 1990b]. The second is a preliminary version of [Grosof and Russell, 1990].

[Russell and Grosof, 1990a] Stuart J. Russell and Benjamin N. Grosof. Declarative bias: An overview. In Paul Benjamin, editor, *Conference on Reformulation and Inductive Bias*. Kluwer Academic, 1990. Edited volume based on the Philips Workshop held in Briarcliff, NY, June 8-10, 1988. Also available as IBM Research report RC15023.

[Russell and Grosof, 1990b] Stuart J. Russell and Benjamin N. Grosof. A sketch of autonomous learning using declarative bias. In Pavel Brazdil and Kurt Konolige, editors, *Machine Learning, Meta-Reasoning, and Logics*. Kluwer Academic, 1990. Based on the workshop held in Sesimbra, Portugal, in Feb. 1988. Also available as IBM Research Report RC15020.

[Russell, 1989] Stuart J. Russell. *The Use of Knowledge in Analogy and Induction*. Pitman (London). Available in the Western Hemisphere through Morgan Kaufmann, San Mateo, California, 1989.

[Scutellà, 1990] Maria Grazia Scutellà. A note on Dowling and Gallier's top-down algorithm for propositional horn satisfiability. *Journal of Logic Programming*, 8:265–273, 1990.

[Selman and Kautz, 1989a] B. Selman and H. Kautz. The complexity of model preference default theories. In M. Reinfrank *et al.*, editor, *Proceedings of the Second International Workshop on Non-Monotonic Reasoning*, pages 115–130, Berlin, Germany, 1989. Springer Lecture Notes on Computer Science.

[Selman and Kautz, 1989b] Bart Selman and Henry A. Kautz. Model-preference default theories. Technical Report KRR-TR-89-7, Dept. of Computer Science, Univ. of Toronto, Toronto, Ontario, Canada M5S 1A4, Apr 1989.

[Selman and Levesque, 1989] B. Selman and H. Levesque. The tractability of path-based inheritance. *Proceedings IJCAI-89*, pages 1140–1145, Detroit, MI., 1989.

[Selman and Levesque, 1990] B. Selman and H. Levesque. Abductive and default reasoning: A computational core. *Proceedings of AAAI-90*, pages 343–348, Menlo Park, CA, 1990. AAAI Press / MIT Press.

[Shoham, 1988] Y. Shoham. *Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence*. MIT Press, Cambridge, Mass., 1988.

[Stein, 1989] Lynn A. Stein. Skeptical inheritance: Computing the intersection of credulous extensions. *Proceedings of IJCAI-89*, pages 1153–1158. Morgan Kaufmann, 1989. Held Detroit, MI.

[Stillman, 1990] Jonathan Stillman. It's not my default: The complexity of membership problems in restricted propositional default logics. *Proceedings of AAAI-90*, pages 571–578, Cambridge, Mass., 1990. AAAI Press / MIT Press.

[Stillman, 1992] Jonathan Stillman. The complexity of propositional default logics. *Working Notes of the Fourth International Workshop on Non-Monotonic Reasoning*, pages 231–236, Artificial Intelligence Program, General Electric Research and Development Center, P.O. Box 8, Schenectady, NY 12301, 1992. Available as a report from the author. Held at the Hawk Inn, Plymouth, Vermont, May 28–31, 1992. Co-chaired by David Etherington and Henry Kautz of AT&T Bell Labs. Sponsored in part by AT&T and AAAI.

[Subramanian, 1989] Devika Subramanian. *A Theory of Justified Reformulation.* PhD thesis, Stanford University, Computer Science Department, Stanford, California 94305, March 1989. PhD dissertation, available as Technical Report Number STAN-CS-89-1260. Author's current address is Computer Science Department, Cornell University, Ithaca, NY.

[Touretzky, 1986] D. Touretzky. *The Mathematics of Inheritance Systems.* Pitman, London, 1986.

[Ullman, 1988] Jeffrey D. Ullman. *Principles of Database and Knowledge-Base Systems, volume 1.* Computer Science Press, Rockville, Maryland, 1988.

[Van Gelder, 1988] Allen Van Gelder. Negation as failure using tight derivations for general logic programs. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming.* Morgan Kaufmann, Los Altos, CA., 1988.

[Zadrozny, 1987] W. Zadrozny. A theory of default reasoning. *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*, pages 385–390, Seattle, Washington, 1987.

[Zadrozny, 1988] Wlodek Zadrozny. A three-level theory of reasoning. Unpublished, IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598., 1988.