17
30

Contents

To Appear in: *Proceedings of the 1992 Workshop on Change of Representation and Problem Reformulation*, held Asilomar, California, April 28-May 1, 1992. Chair: Michael Lowry, NASA Ames Research, Moffett Field, California.

Reformulating Non-Monotonic Theories For Inference and Updating

Benjamin N. Grosof
IBM T. J. Watson Research Center
P.O. Box 704, Yorktown Heights, NY 10598
(914) 784-7100 ; Internet: grosof@watson.ibm.com

April 18, 1992

Abstract

We aim to help build programs that do large-scale, expressive non-monotonic reasoning (NMR): especially, *"learning agents"* that store, and revise, a body of conclusions while continually acquiring new, possibly defeasible, premise beliefs. Currently available procedures for forward inference and belief revision are *exhaustive*, and thus impractical: they compute the entire non-monotonic theory, then re-compute from scratch upon updating with new axioms. These methods are thus badly intractable. In most theories of interest, even backward reasoning is combinatoric (at least NP-hard). Here, we give theoretical results for prioritized circumscription that show how to reformulate default theories so as to make forward inference be *selective*, as well as *concurrent*; and to restrict belief revision to a *part* of the theory. We elaborate a detailed divide-and-conquer strategy. We develop concepts of *structure* in NM theories, by showing how to reformulate them in a particular fashion: to be conjunctively *decomposed* into a collection of smaller "part" theories. We identify two well-behaved special cases that are easily recognized in terms of syntactic properties: disjoint appearances of predicates, and disjoint appearances of individuals (terms). As part of this, we also *definitionally* reformulate the global axioms, one by one, in addition to applying decomposition. We identify a broad class of prioritized default theories, *generalizing default inheritance*, for which our results especially bear fruit. For this *asocially monadic* class, decomposition permits reasoning to be *localized* to individuals (ground terms), and reduced to *propositional*. Our reformulation methods are implementable in *polynomial time*, and apply to several other NM formalisms beyond circumscription.

1 Introduction

Large-Scale, Expressively Rich, Learning Agents: We aim in this work[1] to help build agents that do large scale, expressive non-monotonic reasoning (NMR). We are interested especially in what we call *learning agents*: automatic programs that store, and revise, a body of conclusions while continually acquiring new, possibly defeasible, premise beliefs.

In many applications, information about which defaults take precedence over others (have greater prioritization) is important and available. [2] Many applications need the ability to express fairly arbitrary first-order forms of default beliefs (e.g., induction, law, natural language, communication), as well as fairly arbitrary (finite) partial orders of precedence (e.g., specificity, reliability, and authority are not "layered" (a.k.a. "stratified"))). [Grosof, 1991] defines and discusses the importance of non-layered priority. Non-layered priority is needed, for example, to adequately represent *default inheritance*.

In these applications, we regard as desirable for many reasons, especially validation (both intuitive and rigorous), that a NM formalism be "expressively rich" not only in the above senses, but also that it be equipped with a relatively strong model-theoretic semantics (e.g., cf. Default Logic [Reiter, 1980], circumscription [McCarthy, 1986] [Lifschitz, 1984], and Autoepistemic Logic [Moore, 1985]). In this connection, we also are interested in skeptical or cautious, rather than credulous or brave, entailment.

Current Incapabilities: Currently, expressively rich NMR[3] has found virtually no application on a large scale (more than order of ten defaults), except for the rather special cases of Prolog-style logic programs and simple inheritance cf. AI frame-based systems.

Part of the problem is that there do not yet exist practical inference mechanisms to support storing and revising a limited body of conclusions as a working theory. Currently, for expressively rich NMR [4], the only procedures

[1]part of forthcoming PhD dissertation [Grosof, 1992b]

[2]Note, however, that most of the discussion and results, e.g., about disjoint describability and definitional reformulation and asocially monadic theories, in this paper also apply to the basic case where there are only two "priority levels": for-sure and defeasible.

[3]In [Grosof, 1992b], we make this more precise; here, let us just consider circumscription, Default Logic, and Autoepistemic Logic.

[4]including even the propositional special case and the special case of stratified logic programs with negation [Lifschitz, 1987] [Przymusinski, 1988]

for *forward*[5] inference are *exhaustive*: they compute the entire non-monotonic theory (or, even worse, all credulous extensions). Also, currently, there are no procedures for performing *belief revision* on a body of conclusions, upon receiving new, asserted axioms (an update), beyond the exhaustive method of re-computing everything from scratch. (By "axiom", we mean a premise belief.) [6] By belief revision, we mean modifying the stored conclusions to retract those that are no longer entailed by the newly augmented axiom set.[7] By updating, we mean belief revision plus possibly the inference and storing of some additional conclusions.

Strategy and Summary: In this work, we attack these problems at the level of logical understanding (rather than, say, domain-dependent control of reasoning). Our analytic perspective is that a prime underlying difficulty in the tasks of inference and updating, as well as in specification, is the *logical globality* of NMR: in general, conclusions depend on the *whole* of the axiom set. The exhaustiveness of current methods is, in effect, a manifestation of their caution in dealing with (conflicting) interaction.

We define the concept of a *prioritized database*, using circumscription, as the logical representation of a learning agent that performs *sound*, but incomplete, expressively rich NMR. By database, we mean a subset of a (NM) theory. Prioritized circumscription meets our prime expressive concerns, offers mathematical convenience, and has inference procedures currently available.

We elaborate a detailed "divide and conquer" strategy. We develop concepts of, and results about, *structure* in prioritized circumscriptive theories, by showing how to reformulate them in a particular fashion: to be *conjunctively "decomposed"* hierarchically into a collection of smaller "part" theories, i.e., sub-theories which we call *slices*. We show that it is possible, and useful, to slice within slices. In this way, we map groups of axioms to groups of conclusions. We use the decompositions to analyze the interaction between defaults / parts in a NM theory. Much technical difficulty and trickiness arises from the expressive need to consider non-layered prioritization.

We give theorems that *localize* entailment and thus show how to make forward inference be *selective*, as well as *concurrent*. Exhaustive inference on a slice generates only a part of the global theory. Inferences *within* each

[5]bottom-up. By "backward", we mean totally goal-directed cf. query-answering.

[6]NM formalisms, e.g., JTMS's [Doyle, 1979], having such procedures lack our desired expressive properties.

[7]For simplicity, we assume that these are the only ones removed from storage.

slice (sub-theory) can be performed in parallel with inference *within* every other slice. All non-monotonic inference can be localized to the slices; only *monotonic* inference is required between the slices. We give theorems that localize retraction and thus show how to make belief revision be *partial* in the sense that, for a given update, the arena of potential retraction is known to be restricted to a particular part of the previous database.

Our results enable the exploitation of other results on inference and belief revision that are limited to expressive special cases, say to do exhaustive forward inference in polynomial time (e.g., the "sympathetic-solitary" case in [Grosof, 1992b] that generalizes predicate completion [Clark, 1978] and the Closed World Assumption). These special case results can be applied to one, or several, slices, even when they do not apply to the global theory.

Our results are about well-behaved special cases that are easily recognized in terms of syntactic properties. The first *"cleanly slice-able"* property is disjointness of mentioned predicates. We show that if the for-sure and default axioms can be partitioned into groups which are disjoint in terms of the predicate symbols they mention, then non-monotonic inference based on each partition can proceed without considering the axioms in the other partitions: those other axioms are irrelevant in an important sense, as far as that partition is concerned. We show this implies that updating with new for-sure and default axioms that span only some of the previous partitions does not require retracting previous conclusions based purely on the remaining partitions: they are *safe*.

Most large practical applications, however, do not display such perfect partitionability of mentioned predicates. The real power from our result about disjointness of predicates comes when it is *combined* with another kind of reformulation: of the axioms in a given global axiom set, not just of the global axiom set into decomposed constituent axiom sets. We define a concept of *disjoint describability*: syntactic partitionability *after* definitional reformulation of the axioms. As part of this, we give a logical *definition* of a particular kind of definitional (i.e., equivalence-preserving) reformulation with respect to a background theory, modifying the standard logical idea of a conservative extension. We also discuss, and use, another kind of reformulation: to break up open defaults (i.e., schema-type, as opposed to closed, i.e., propositional) into cases. An important difference from definitionally reformulating monotonic theories is that two default axioms $D1$ and $D2$ cannot, in general, be equivalently replaced by the default axiom corresponding to the conjunction

$D1 \wedge D2$ the way that two for-sure axioms can always be equivalently replaced by their conjunction $B1 \wedge B2$. This is why we need to consider reformulation of the axioms one-by-one.

Using these definitional and default-cases reformulations, we arrive at our second cleanly slice-able, yet syntactically recognizable, property: disjointness of mentioned individuals. We show that a fairly broad class (*"asocially monadic"*) of prioritized default circumscriptions is cleanly slice-able into one slice theory per named individual (ground term in the language) plus a remainder-case slice. Each of these individual-wise slices is propositional, and is, essentially, much simpler than the global, in several ways: number of axiom instances (especially, potential primitive default conclusions), availability of inference methods, and availability of known computational complexity results. (Unfortunately, we do not have space to discuss NM inference methods and complexity results in detail. But see the final section.) The asocially monadic class **includes, as a special case, default inheritance networks** of the kind studied by [Touretzky, 1986] and used in many AI frame-based systems. The asocially monadic class is more general, however: it permits more than one antecedent in default rules, free use of negation, and freer use of disjunction.

While definitional reformulation is hard in general, we have a **polynomial-time algorithm** (omitted in this draft to save space and preserve focus) to peform the recognition and exploitation of this asocial-monadic reformulation and decomposition. More precisely, the algorithm is $O(n^3)$, where n is the size of the (global) axiom set (which is, moreover, typically much smaller than the whole theory, of course).

We show that our conjunctive decomposition results imply safeties in belief revision. We illustrate the problems of scale in learning agents with an extended example of a prioritized database and show that our safety theorems capture much of the preponderant stability (i.e., most beliefs are preserved after each update) that this database displays through its sequence of updates. We show, using the example, that decompositions on these two bases combine *synergistically*, as well as *hierarchically*: it is useful to slice within slices.

Finally, we observe that our formal reformulation methods are implementable at reasonable cost, and apply to several other NM formalisms. We have **polynomial-time algorithms** (again, omitted here due to space and focus) for disjoint predicates, as well as for asocially monadic, also in $O(n^3)$, where n is the size of the (global) axiom set.

2 A Motivating Example

Next, we give an extended example of a learning agent, in the domain of common-sense default reasoning, that illustrates issues of selective forward inference and partial belief revision on a large scale. We present it first at an intuitive level, and formalize it later.

We adopt the following notation. A $\bullet\!>$ prefix indicates that the sentence that follows is a *base* axiom, i.e., has for-sure (non-defeasible) belief status. A $:>$ prefix indicates that the formula that follows is a *default* axiom (roughly, a normal default without pre-requisite in Default Logic). Its label, e.g. $(d1)$, serves as a tag for defining prioritization-type precedence between defaults via $\mathcal{PREFER}$ (*prioritization*) axioms. These define a strict partial order of precedence, via transitive closure. $\mathcal{PREFER}(d1, d2)$, for example, means that the default axiom with label $(d1)$ has strictly greater precedence (priority) than the default axiom with label $(d2)$.

We make the Uniqueness of Names Assumption (consider it included as a for-sure axiom). 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.

In this example, the agent starts with no beliefs, then accumulates axioms by receiving updates. After each update, the agent draws a bunch of conclusions (say, ground first-order sentences), both monotonically and non-monotonically, and retracts some of its previous conclusions. Each $\mathcal{U}_i$ indicates an update, consisting of one or more axioms. Axioms are numbered. In addition, we show explicitly with $\mid\!\approx$ and $\not\mid\!\approx$ a few of the more interesting NM conclusions and retractions, respectively, about which discussion will revolve. Note that, by "conclusion", we always mean in the skeptical sense.

The first update consists of a default axiom, that bats have two legs, together with some for-sure axioms. Non-monotonic (default) conclusions include that known bats are two-legged. The second update consists of another, default axiom, that mammals have four legs, together with the precedence axiom, that this new default has lower priority than the previous, more specific one. The third update consists of two default axioms about emergency disaster situations, plus some associated for-sure information. Intuitively, since the axioms in this new update are about a totally different topic than the previous axioms, they should not result in having to retract any of the previous conclusions. Moreover, intuitively, the agent should be able to draw the conclusions from these new axioms without even having to consider the previous ones in detail.

The fourth update consists of some for-sure information about two named individuals, *Joe* and *Spot*, that violates some previous default conclusions. Intuitively, since there is no information that "connects" any other named individuals to *Joe* and *Spot*, these new axioms should not result in having to retract any of the previous conclusions that are *not* about those named individuals: e.g., that are about some other named individuals. For example, the previous default conclusion $2legs(Betsy)$ should not have to be retracted.

Later, we will show how to capture these intuitions as formal guarantees.

3 Formal Definitions: Prioritized Circumscription

We **define** our notation for axioms from section 2 as a meta-language (the Circumscriptive Language of Defaults, or CLD for short) that, at any point in the update sequence, specifies a *prioritized "default" circumscription* of the form:

$$PDC(B; D; R; fix\, W; Z) \stackrel{\text{def}}{\equiv}$$
$$B[Z] \wedge \neg\exists Z'.\ B[Z'] \wedge Z \prec_{(D;R)} Z' \wedge W = W'$$

Here, B is the conjunction of the sentence parts of all of the for-sure axioms. D is the tuple of the default axioms' formula parts. R is a strict partial order of precedence (priority). It is the transitive closure of the precedence relation specified by the pairwise comparisons in the $\mathcal{PREFER}$ axioms. Its domain, accordingly, is the set of default axiom labels. Z is the tuple of all mentioned predicate symbols; e.g., in the example, $\langle bat, dog, mammal, 2legs, 4legs, fire, \ldots\rangle$. $W \subset Z$ is the tuple of predicates that are *fixed*. Fixing is a standard notion in the circumscription and non-monotonic reasoning literature. Fixing is part of the specification of non-monotonic reasoning. Intuitively, fixing some symbols implies that any formula that mentions only those symbols is *immune* to the circumscription operation in the sense that it can be concluded non-monotonically, i.e., from the circumscription, only if it can be concluded "monotonically", i.e., from the for-sure axioms B alone. For simplicity, we also *fix* (do not vary and second-order quantify over) all function symbols.

This assumption can easily be relaxed. This assumption is typical in the circumscription literature. Uniqueness of Names, plus Domain Closure, implies that functions are effectively fixed, for example. For the sake of simplicity, in this paper, we for the most part do not consider fixing of predicates, only of functions: W is empty. We omit further details about fixing to save space and to preserve focus; see [Grosof, 1992b] for more.

Prioritized default circumscription is a slight generalization of prioritized predicate circumscription cf. [Grosof, 1991]. We employ it and CLD to clarify the definitions of axiom sets and of updating, and the intuitive relationship to other formalisms for default reasoning. [Grosof, 1992b] shows as a **theorem** the equivalence of any prioritized default circumscription to a corresponding, abnormality-style, prioritized predicate circumscription, generalizing a previous result that appeared in [Lifschitz, 1984]. Note that our definition can express minimizing predicates as a special case: e.g., $:> ab_i(x)$, where ab_i is an abnormality predicate.

We let N stand for the index tuple of D: it is just (isomorphic to) the tuple of the labels of the default axioms. I.e., in the example, after the second update, the elements of $D[Z]$ are:

$$\lambda x.\ bat(x) \supset legs2(x)\ ,$$
$$\lambda x.\ mammal(x) \supset 4legs(x)$$

and $N = \langle d1, d2 \rangle$. $R(j,i)$ means that the default with label j has strictly higher priority than the default with label i. $\prec_{(D;R)}$ is defined as the strict version $(\preceq_{(D;R)} \wedge \neg \succeq_{(D;R)})$ of the prioritized "formula" pre-order $\preceq_{(D;R)}$:

$$Z \preceq_{(D;R)} Z' \ \stackrel{\text{def}}{\equiv}$$
$$\forall i \in N.\ [\forall j \in N.\ R(j,i) \supset$$
$$(\forall x.\ Dj[Z,x] \equiv Dj[Z',x])]$$
$$\supset\ (\forall x.\ Di[Z,x] \supset Di[Z',x])$$

Here Dj and Di refer to the j^{th} and i^{th} members, respectively, of the tuple D.[8] We **define** the corresponding circumscriptive prioritized default *theory* as the set of all conclusions entailed (model-theoretically, in second-order logic) by the prioritized default circumscription. [9] [10] We **define** a *prioritized database*

[8]For notational simplicity, we ignore the potentially different arities of the various open formulas Di.

[9]See [Grosof, 1991] and [Grosof, 1992b] for more discussion of how prioritized circumscriptions are defined. Note that the prioritization p.o. R is not necessarily layered (stratified) (indeed, in our example, it is not) as it was in [Lifschitz, 1985].

[10]In section 5, we generalize the definition above to include the explicit "fixing" of a set

(PDB) to be a pair, consisting of: a CLD axiom set $\mathcal{A}$ (in the example, the current collection of the updates ($\mathcal{U}_i$'s)); and an associated *prioritized database theory* $\mathcal{DB}$, which is some subset of the prioritized default circumscriptive theory $\mathcal{C}(\mathcal{A})$ specified by $\mathcal{A}$. Here, $\mathcal{C}$ is the non-monotonic theory operator for the CLD formalism.

4 Decomposition: Concepts

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: 1) to define safe versus unsafe zones for belief revision; 2) to define relevant versus irrelevant context for inference (and for specification); and 3) to define the structure and organization of an overall ("global") prioritized database. In classical logic, we take for granted such an idea of a part of theory. However, the dependence of entailment on, in general, the entire *global* axiom set means that we have to "work for it" in NM logical systems.

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 corresponding sub-theories (the $\mathcal{ST}_i$'s), where each sub-theory is the result of applying $\mathcal{C}$ to a constituent axiom set: $\mathcal{ST}_i \stackrel{\text{def}}{=} \mathcal{C}(\mathcal{SA}_i)$.

In CLD, we **define** $\mathcal{T}$ to be the result of *conjunctive* combination when $\mathcal{T}$ is $Cn(\bigcup_{i=1,\ldots,n} \mathcal{ST}_i)$; where Cn is the *monotonic* consequence (theory) operator in classical logic. When the corresponding axiom sets are understood, we will say that the global *theory* is conjunctively decomposable into these *slice* sub-theories. [11]

of formulas, e.g., a subset of the predicates. [Grosof, 1992b] gives details.

[11] *Serial* combination 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 can be described in this manner. Details about serial decomposition are omitted due to considerations of space and focus. See [Grosof, 1992b] for more.

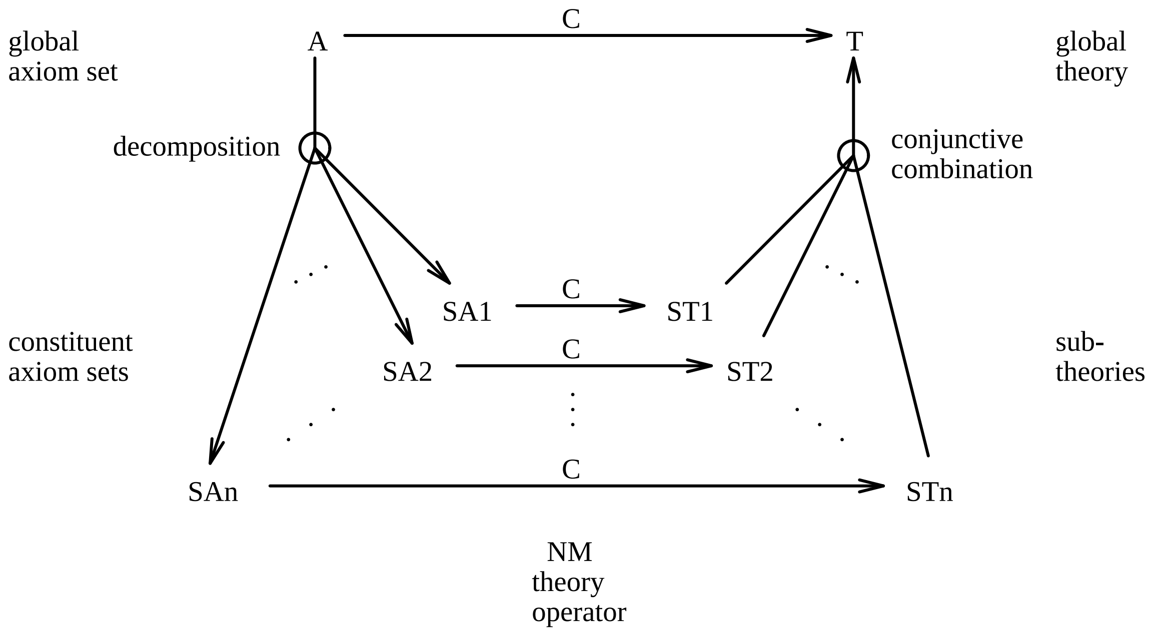

Figure 1: Conjunctive Decomposition: a conceptual flow diagram. A global theory $\mathcal{T}$ can be obtained *either* directly by applying the NM theory operator $\mathcal{C}$ to the global axiom set $\mathcal{A}$, *or* indirectly (but equivalently) via decomposition. In decomposition, the global axiom 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: $\mathcal{ST}_i \stackrel{\text{def}}{=} \mathcal{C}(\mathcal{SA}_i)$.

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 $n = 2$:

$$PDC(B; D; R; Z) \equiv$$
$$PDC(SB1; SD1; SR1; Z) \wedge$$
$$PDC(SB2; SD2; SR2; Z)$$

Figure 1 illustrates conjunctive decomposition with a flow diagram.

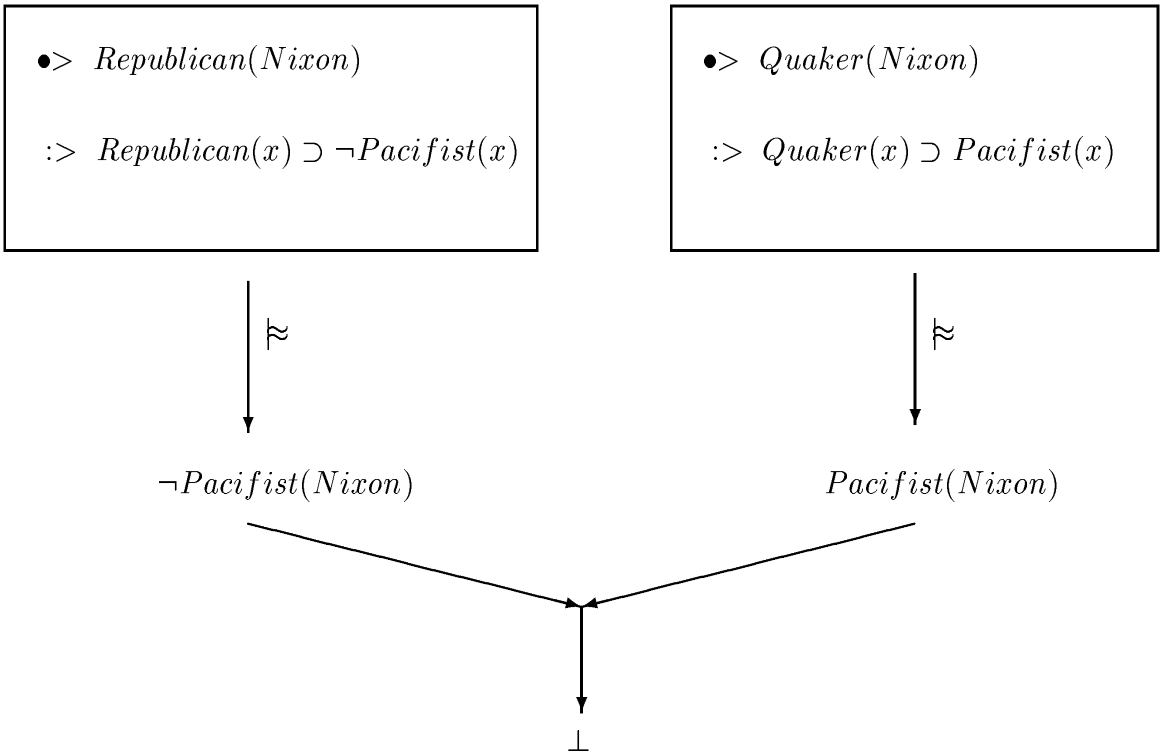

Figure 2: Non-modularity: Quakers and Republicans. (Default axiom labels not shown.)

Conjunctive decomposition is thus a kind of **reformulation** *or representation change.* The global axiom set and theory $\langle A, T\rangle$ are transformed into a collection of constituent axiom sets and slice sub-theories: $\langle\langle \mathcal{SA}_1, \mathcal{ST}_1\rangle, \ldots, \langle \mathcal{SA}_n, \mathcal{ST}_n\rangle\rangle$.

Most Subsets Do Not Qualify As Constituents for Decomposition: Note that, in general, in non-monotonic reasoning, one cannot blithely partition a global axiom set into a bunch of (distinct, or, more generally, overlapping) subsets (whose union is the global axiom set) any old way and get a conjunctive decomposition. This is because the axioms in one subset may conflict with those in another.

E.g., consider the classic Quaker-Republican example of conflict in default reasoning: there are two default axioms, one saying that Quakers are typically Pacifists, and another saying that Republicans are typically non-Pacifists. In addition, there are two for-sure axioms: that Nixon is a Quaker, and that he is a Republican. Suppose we consider two subsets: one containing the Quaker axioms, and another containing the Republican axioms. Treating a subset as a constituent axiom set means drawing non-monotonic conclusions from it as if there were no other axioms around. Doing so, from the first (with Quaker) one gets the default conclusion that Nixon is a Pacifist; from the second, one gets the default conclusion that Nixon is a non-Pacifist. Taking the conjunction of these two "sub-theories" thus results in garbage: inconsistency. Yet the actual global theory is consistent: neither conclusion about Pacifism is sanctioned. Figure 2 illustrates.

Our perspective is that, in general, non-monotonicity means a kind of **logical non-modularity**: when attempting to draw conclusions from a subset of the global axiom set, one must keep in mind the context of the remainder of the global axiom set. If one considers that remainder as an "internal" update, then that update may be non-monotonic. Another way to view this situation is that non-monotonicity means **logical globality**: in general, a non-monotonic conclusion cannot be drawn until the *entirety* of the global axiom set is considered.

Locality:
Suppose we can find a conjunctive decomposition in which for some i, the slice's axiom set is 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. Then we know that all the remaining axioms $(\mathcal{A} - \mathcal{SA}_i)$ in the global axiom set are *irrelevant context*, in an important sense, relative to the slice's axiom set $\mathcal{SA}_i$. In this case, 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 generally for the NM formalism. 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 $\mathcal{C}(\mathcal{SA}_i)$. By "irrelevant" above, then, we mean that one does not need to consider the remainder of the global axioms in order to do the essential non-monotonic aspect of the reasoning from $\mathcal{SA}_i$.

In the rest of this paper, we will be only considering decompositions that are clean. ([Grosof, 1992b], however, discusses the usefulness of decompositions that are not clean, e.g., decompositions on the basis of higher versus lower priority.)

Observe that in clean slicing, the constituent axioms sets are each smaller, and thus simpler, than the global axiom set. In prioritized default circumscription, and in other expressively rich NM formalisms, the computational complexity of non-monotonic reasoning (including, full forward inference and belief revision) is worse than monotonic reasoning. Non-monotonic reasoning (full forward inference and belief revision) in each slice, and via monotonic conjunctive combination, is thus **computationally less complex** than non-monotonic reasoning in the global theory.

Partitioning Axioms As Kind of Reformulation:
Our perspective, therefore, is that, in non-monotonic reasoning, decomposing, e.g., partitioning (see Theorems 1 and 12), a global axiom set into constituent axiom sets is a quite non-trivial kind of reformulation. This is very different from the situation in classical monotonic reasoning.

Safeties of Updating:
Suppose, in a conjunctive decomposition, that $\{\mathcal{SA}_1, \ldots, \mathcal{SA}_k\}$ are present both before and after an update $\mathcal{U}$. I.e., suppose that some of the constituent axiom sets in a decomposition after an update $\mathcal{U}$ are unchanged from (i.e., are the same as) in a decompositon before that update. Then we know that all of the conclusions in the conjunctive combination of their associated slices are safe under the update.

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'. Thus we can often choose grain size hierarchically during conjunctive decomposition.

Sequencing of Inference: See section 1 about concurrency.

5 Disjoint Predicates

Our results will all make use of the following idea of decomposing the specified prioritization.

Composing Prioritization:
The concept of prioritization over groups of defaults is natural in the specification process for many applications: often a group of defaults corresponds to a topic. [Grosof, 1991] introduced, and [Grosof, 1992b] elaborates, this idea of *"composing"* prioritization, in which an overall prioritization p.o. R over

the domain of individual defaults is equivalent to the result of composing an *external* prioritization p.o. RE, defined over groups, with a tuple RI of prioritization p.o.'s, one (RIi) per group, that each represent the prioritization *internal* to that group: $R = RE{*}RI$. Groups may, in turn, be composed of groups. Thus we may define prioritizations of prioritizations, in hierarchical or recursive fashion. Our example displays this structure.

Our first result is about decomposition on the basis of syntactic disjointness of predicates. It captures a basic case of the intuition that syntactically "having nothing to do with each other" should imply strong irrelevance of the kind we discussed in the last section.

Theorem 1

(Clean Decomposition, given Disjoint Predicates)
Let $PDC(B; D; R; Z)$ be a global PDC. Let $\{B1[Z1], \ldots, Bk[Zk]\}$ be a partition of the base axioms $B[Z]$, and let $\{D1[Z1], \ldots, Dk[Zk]\}$ be a partition of the default formulas $D[Z]$, where the predicate tuples $Z1, \ldots, Zk$ are a (disjoint) partition of Z. I.e., in terms of CLD, let there be a partition, of the base and default axioms, where the predicates mentioned in each element of the partition are disjoint. If a certain condition (0) (see below) on the prioritization R is satisfied, then

$$PDC(B; D; R; Z) \equiv \bigwedge_{j=1}^{k} PDC(Bj; Dj; RIj; Z)$$

(Note that the Z on the right-hand side can be equivalently replaced by Zj.) Condition (0) is defined as: either, R is the composition of some prioritization RE with the tuple RI of the internal prioritizations of each partition; or, R is layered (stratified). The composition condition for non-layered R corresponds, intuitively, to a kind of a partitionability of the prioritization. Note the special case of empty R satisfies (0).

Proof Overview: Surprisingly non-trivial. The essence is to use the ability to separate existential quantifiers in the right-hand-side part of the circumscription formula (cf. section 3). Non-layered prioritization makes this tricky: hence the prioritization conditions in the theorem. □

In terms of CLD, Theorem 1 tells us that syntactic disjointness implies *irrelevance* in the sense that we discussed in the last section; the decomposition by syntactic partition is a clean slicing.

Theorem 1 immediately yields a powerful result about inference.

Theorem 2

(Locality of Inference, given Disjoint Predicates)
In Theorem 1, each slice j is sound and complete, relative to the global theory, for inference over its corresponding sub-language (partition of the predicates). That sub-language consists of the formulas that mention only the predicates Zj. This locality holds both for forward inference, and for backward inference (query-answering). Note that to perform inference using any subset Y of the predicates Z, one need only work in the conjunctive combination of those slices whose predicates cover that subset Y.

Theorem 1 also immediately yields a powerful result about belief revision.

Theorem 3

(Safety of Updating, given Disjoint Sub-Languages)
In CLD, let the previous axiom set be partitionable according to Theorem 1. Let an update $\mathcal{U}$ consist of base, default, and prioritization axioms, such that the formula parts of the base and default axioms mention only predicates from a (possibly empty) subset of the previous partitions, and such that the global prioritization condition (0) is still met. Then all of the previous conclusions derived solely from the rest of the partitions' slices do not require retraction.

Application to Main Example:
The above theorems capture the first intuition that we discussed in section 2. At each point in the sequence of updates, Theorem 2 implies that inference can be localized: inferences about legged-ness can be performed in the slice that contains only the axioms about legged-ness, and likewise for meetings. Figure 3 illustrates the conjunctive decomposition cf. Theorem 1 after the last update. Theorem 3 guarantees that after each meetings update, all of the previous conclusions drawn from the legged-ness slice are safe, and vice versa.

Generalizations:
Theorems 1, 2, and 3 generalize in several directions. Firstly, predicate (and function) symbols may overlap between the constituent axiom sets as long as they are fixed in the circumscription (see earlier discussion about fixing in section 3). Intuitively, it is OK to specify some predicate (and function) symbols as fixed if it is OK not to infer any default conclusions expressible

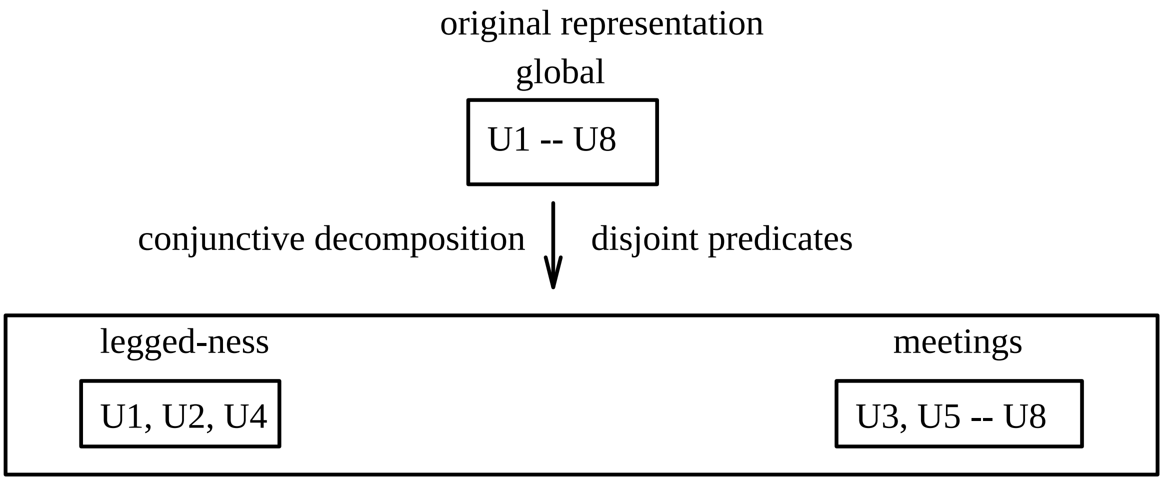

Figure 3: Conjunctive Decomposition using Disjoint Predicates: In our main motivating example (section 2), we conjunctively decompose the global axiom set (after the last update $\mathcal{U}_8$) into two slices by employing the disjoint predicates result (Theorem 1): one slice about legged-ness, and the other slice about meetings. In the bottom half, each inner box stands for a constituent axiom set.

purely in terms of those symbols. Secondly, the prioritization condition can be relaxed somewhat.

Definitional Reformulation of Axioms:
Thirdly, and most interestingly (see discussion toward end of section 1 about source of power), one can decompose with irrelevance (slice cleanly) as long as one can definitionally *reformulate* the global axiom set to meet the partitionability condition. (See Theorem 12.) One interesting such case is reasoning about one individual object, e.g., Joe in our example, at a time. (See Theorem 16.) Often (e.g., for the legged-ness axioms in our example), such re-formulability is easily (time polynomial in the number of axioms) detectable syntactically. We pursue all this in the next two sections.

6 Basic Definitional Reformulation of Axioms, One-by-One

Next, we define a particular kind of definitional reformulation. This kind of reformulation maps each formula in one formulation into a correspondent formula in another formulation, while preserving equivalence, i.e., without loss of information. Our motivation for considering this limited kind of reformulation is our intended application: to disjoint describability and its asocial-monadic special case. Why do we do the reformulation "one axiom at a time", i.e., *one-by-one*? Much of the reason is that there is an important difference between default / NM reasoning and monotonic reasoning.

We take for granted in monotonic logics that a collection of for-sure (base) axioms $B1, \ldots, Bm$ can be equivalently replaced by the axiom $B1 \wedge \ldots \wedge Bm$. In prioritized default circumscription and most other expressively rich NM formalisms, however, one cannot, in general, equivalently replace the pair of default axioms (whose default formulas are) $D1$ and $D2$ by the default axiom (whose default formula is the conjunction) $D1 \wedge D2$ (even in the case without priorities, i.e., when the prioritization is empty). Informational "grain size" of the defaults is important: having the two separate defaults means that, for example, $D2$ may "succeed" (i.e., be concluded non-monotonically from the defaults) even if $D1$ is "defeated" (e.g., is violated by the for-sure information), unlike if the only default present is $D1 \wedge D2$. We will need equivalence-preserving (and information-preserving) reformulation in order to apply the decomposition on the reformulated representation back onto the original representation.

Circumscription is defined in terms of second-order logic. We thus find it convenient and natural to define the kind of definitional reformulation we will need in terms of second-order logic, as well. We build on the standard idea of a conservative extension, drawn from the classical logical literature. In this and the next section, we then develop several, increasingly complex notions of definitional reformulations, in order to handle the grouping structure in various stages of our reformulations: groups of predicates, groups of individuals, groups of formulas.

In this paper, we mainly address reformulations oriented around disjointness of mentioned (predicate and function) symbols. It is thus convenient to define our changes of representation in terms of changes in the symbols mentioned.

First Cut at Definitional Reformulation:
What does it mean to definitionally reformulate theories (or formulas) while preserving equivalence? At first glance, it simply means to introduce some definitions (of new symbols) which logically imply (entail) the equivalence of a theory expressed in an original set of symbols (i.e., an original representation) to a new theory expressed in those new symbols (i.e., a new representation). E.g., let $A1[P]$ be the original theory, let $U[P,Q]$ be some definitions of new symbols Q in terms of the old symbols P, e.g., a conjunction of explicit definitions:

$$U[P,Q] \stackrel{\text{def}}{\equiv}$$
$$(Q1 \equiv E1[P]) \wedge \ \ldots \ \wedge (Qm \equiv Em[P])$$

(where m is the length of the tuple Q), and let $A2[Q]$ be a new theory that is equivalent to $A1[P]$ given $U[P,Q]$:

$$U[P,Q] \ \models \ A1[P] \equiv A2[Q]$$

More generally, we can permit the new representation to use some of the old symbols; let W be the overlap symbols between the old and the new. Suppose $A1[W,Y]$ is the original theory, $A2[W,Y,Z]$ is the new theory, and $U[W,Y,Z]$ is the (conjunction of) definitions of new symbols Z in terms of W and Y, e.g.,

$$U[W,Y,Z] \stackrel{\text{def}}{\equiv}$$
$$(Z1 \equiv E1[W,Y]) \wedge \ \ldots \ \wedge (Zm \equiv Em[W,Y])$$

(where m is the length of the tuple Z); and suppose

$$U[W,Y,Z] \ \models \ A1[W,Y] \equiv A2[W,Z]$$

Then we call U a *"putative"* definitional *reformulator*.

Observation 4

(Subtlety: Uninformativeness and Consistency)
However, there is a subtlety. To us, part of the intuition behind the idea of an definitional reformulation is that the equivalence is non-spurious, i.e., that the definitions themselves are not introducing information. Unfortunately, merely requiring U to be a conjunction of explicit definitions allows spuriosity and informativeness.

Consider the following **example**. Let W be empty. Let $Y \stackrel{\text{def}}{\equiv} \langle Y1, Y2 \rangle$, where $Y1$ and $Y2$ are 0-ary predicates.[12] Let $A1[W,Y]$ be defined as $Y1 \wedge \neg Y1$.

[12]We do not use $Y2$ immediately, but we will use it later when we continue this example

Let the definitions U be $(Z1 \equiv Y1) \ \wedge \ (Z2 \equiv \neg Y1)$, where $Z1$ and $Z2$ are 0-ary predicates. Let $Z \ \stackrel{\mathrm{def}}{=} \ \langle Z1, Z2 \rangle$, and let $A2[W, Z]$ be defined as $Z1 \wedge Z2$. Then U implies that $A1$ is equivalent to $A2$. Yet this contravenes our intuition of a reasonable definitional reformulation. $A1$ is inconsistent, i.e., is equivalent to $False$. $A2$, by contrast, is consistent.

Viewing the direction of reformulation from $A2$ to $A1$, in effect U is introducing some information, namely that $Z1 \equiv \neg Z2$. The source of this problem is that, even though U is a conjunction of explicit definitions, U is itself not always consistent when it is viewed in this "return direction" of the reformulation (i.e., from $A2$ to $A1$). Yet, to us, any notion of equivalence-preserving definitional reformulation ought to be *symmetric*, i.e., kosher in *both* directions: from $A1$ to $A2$ *and* from $A2$ to $A1$. We would, therefore, like to impose some kind of additional constraint on U to guarantee intuitive uninformativeness and non-spuriosity of the equivalence between the two representations. Below, we do this by formalizing U's consistency and its relationship to directionality more precisely.

The idea of a conservative extension, standard in the classical logical literature, provides a nice notion of uninformativeness in terms of mentioned symbols.

Definition 5 (Conservative Extension)

Let $A1[P]$ be a formula[13] 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:

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

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

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:

in the discussion after Definition 7.

[13]in (higher-order) classical logic

$$A2[P,Q] \models G[P] \quad \Longleftrightarrow \quad A1[P] \models G[P]$$

Suppose that

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

Then we say that $U[P,Q]$ is a *conservatively extending update* to $A1[P]$.

Intuitively, we can thus view a conservatively extending update $U[P,Q]$ as uninformative in a precise sense, namely about the old symbols P.

Notation:
Let $D \leq E$ stand for the universally quantified implication $\forall x.\ D(x) \supset E(x)$, where D and E are open formulas with the same arity of free variables (i.e., are similar), and x stands for a tuple of free individual (object) variables. Let $D{=}E$ then be defined analogously as the universally quantified equivalence $\forall x.\ D(x) \equiv E(x)$. We also apply this notation to tuples $D = \langle D_1, \ldots, D_m \rangle$ and $E = \langle E_1, \ldots, E_m \rangle$: e.g., $D{=}E$ stands for

$$D_1{=}E_1 \wedge \ldots \wedge D_m{=}E_m \ .$$

Fact 6

(Explicit Definitions Are Conservative)
(Conjunctions of) explicit definitions of new symbols (e.g., predicates) are always conservatively extending updates. I.e., in Definition 5, suppose $U[P,Q]$ is a conjunction of explicit definitions of each symbol in Q:

$$Q \;=\; E[P]$$

(Here, we are using the tuple $=$ notation introduced above, and applying it also to functions and terms.) Then $U[P,Q]$ is a conservatively extending update, for any $A1[P]$.

Conservative Extension, Uninformativeness, and Directionality:
Equipped with the idea of a conservative extension, we are now ready to return to the question of refining the basic idea of definitional reformulation. In our "first cut" above, we found a need to formalize the constraint that the putative definitional reformulator U be uninformative, in both directions of the reformulation. In Definition 5, we observed that the property that a "definitional" reformulator $U[P,Q]$ is a conservatively extending update precisely expresseses U's uninformativeness, in the direction of $A1$ to $A2$, i.e., about P. There, however, U is not really quite a reformulator in the sense we discussed in the "first cut", since $A2$ mentions not just the new symbols Q, but also

the old symbols P. However, we can *extract* the notion of uninformativeness present there, i.e., the "conservatism" in the idea of a conservative extension.

The property that U is a conservatively extending update is: $A1[P] \models \exists Q.\, U[P,Q]$ which we can also write as:

$$\models \; (\forall P.\, A1[P] \supset \exists Q.\, U[P,Q])$$

One can view the right-hand-side as a satisfiability (i.e., consistency) property. This satifiability / consistency is conditional on $A1$.

We take this conservativeness property as the basis for uninformativeness of a (putative) definitional reformulator U. However, we need the "return direction" uninformativeness as well:

$$A2[Q] \models \exists P.\, U[P,Q]$$

which we can also write as:

$$\models \; (\forall Q.\, A2[Q] \supset \exists P.\, U[P,Q])$$

Definition 7

(Definitional Reformulator — Basic Case)
We say that $U[W,Y,Z]$ is a *definitional reformulator* (basic case) between two formulas $A1[W,Y]$ and $A2[W,Z]$ (where W, Y, and Z are distinct tuples of symbols) when:

1. U implies the equivalence of $A1$ and $A2$:
$$\models \; U[W,Y,Z] \supset (A1[W,Y] \equiv A2[W,Z])$$

2. U is uninformative, i.e., conservative, in both directions of the reformulation, i.e., with respect to $A1$ and with respect to $A2$:
$$\models \; (\forall W,Y.\, A1[W,Y] \supset \exists Z.\, U[W,Y,Z])$$
$$\models \; (\forall W,Z.\, A2[W,Z] \supset \exists Y.\, U[W,Y,Z])$$

Discussion; Directionality:
Having the second direction, in addition to the first direction, of the conservativeness property in Definition 7 rules out the nastily-behaved example that we discussed in Observation 4. However, the conservativeness property in Definition 7 reassuringly does permit, for **example**, the following, more intuitively reasonable basic-case definitional reformulator:

$$U[W,Y,Z] \stackrel{\text{def}}{\equiv} (Z1 \equiv Y1) \wedge (Z2 \equiv \neg Y2)$$

(where the symbols are as in the example discussed in Observation 4) for any $A1$, $A2$.

The property that U consists exclusively of (a conjunction of) explicit definitions ensures, in general, only *one* direction of conservativeness.

Conditionality Versus Unconditionality of Conservativeness:
Definition 7 is perhaps too "custom" in one regard, however. The conservativeness property is conditional: it depends on the particular $A1$ and $A2$. This is perhaps unsatisfactory intuitively, at least for some purposes, as a notion of "definitional" in "definitional reformulator".

Alternative Definition of Conservativeness: Unconditional Version:
As an alternative definition of the basic case of definitional reformulator, we **observe** that one can use a stronger (i.e., more strongly constrained, special case) notion of conservativeness instead:

$$\models \ \forall W, Y.\, \exists Z.\, U[W, Y, Z]$$
$$\models \ \forall W, Z.\, \exists Y.\, U[W, Y, Z]$$

to replace the conservativeness property (2.) in Definition 7. This *"unconditional"* version of the conservativness property does *not* depend on $A1$ and $A2$: i.e., it implies that the *"conditional"* conservativeness property (2.) in Definition 7 holds for *any* $A1$ and $A2$.

Alternative Definition of Conservativeness: Backgrounded Version:
As an intermediate position between the conditional and unconditional versions of the conservativeness property, we **observe** that one can formulate conditionality in a somewhat abstracted fashion: in terms of the symbols W that are in common between the two representations. We will find it convenient for our later definitions to employ a notion of a *background* $G[W]$ to the reformulation. One can view $G[W]$ as, in effect, included in both $A1[W, Y]$ and $A2[W, Z]$. We then define the *"backgrounded"* version of the conservativeness property as:

$$G[W] \ \models \ \forall Y.\, \exists Z.\, U[W, Y, Z]$$
$$G[W] \ \models \ \forall Z.\, \exists Y.\, U[W, Y, Z]$$

In the remainder of this paper, we will use this last, "backgrounded" version of the conservativeness property. We do so in order to formally simplify our later definitions of more complex kinds of definitional reformulators and reformulations, which are oriented towards particular uses. However, the "conditional" version of the conservativeness property is more fundamental and general, we believe, and is interesting to explore: we plan to do so in the future.

No Requirement of Explicitness:
Note that in Definition 7, we did *not* require U to be in the form of a conjunction of explicit definitions of new symbols in terms of old symbols. *We formalized / summarized the "definitional" flavor of the reformulator as, simply, its conservativeness.* Our definition of definitional reformulator thus allows U to consist of implicit definitions (e.g., with recursion) and partial definitions (i.e., necessary and sufficient conditions). (Later, in our result about the asocial monadic special case of disjoint describability (Theorem 16), the reformulator will consist exclusively of explicit definitions, however.)

Next, we define a definitional reformulation of a group of formulas, using a single common reformulator: one-by-one, into a new group of formulas. For this purpose, it is convenient to be able to abstract away from conditionalizing conservativeness on each of those formulas: we thus use the backgrounded version of conservativeness.

Definition 8 (Group Reformulator)

Let $ET1[W,Y]$ and $ET2[W,Z]$ each be a similar[14] tuple of formulas; these formulas may be open or closed. We call each tuple a *group*. Let $U[W,Y,Z]$ and $G[W]$ be closed formulas. We say that $U[W,Y,Z]$ is a *group reformulator between* $ET1[W,Y]$ *and* $ET2[W,Z]$*, given the background* $G[W]$ when:

1. U is conservative (given the background) with respect to Y and also with respect to Z:
   $$G[W] \models \forall Y.\, \exists Z.\, U[W,Y,Z]$$
   $$G[W] \models \forall Z.\, \exists Y.\, U[W,Y,Z]$$

2. U reformulates each formula in either group into the corresponding formula in the other group. I.e., U implies the equivalence of corresponding member formulas (subscripted by j) in the two groups:
   $$U[W,Y,Z] \wedge G[W] \models \forall j.\, ET1j[W,Y] = ET2j[W,Z]$$

[14]**Terminology**: By "similar", we mean of same length, and with same arities for their members.

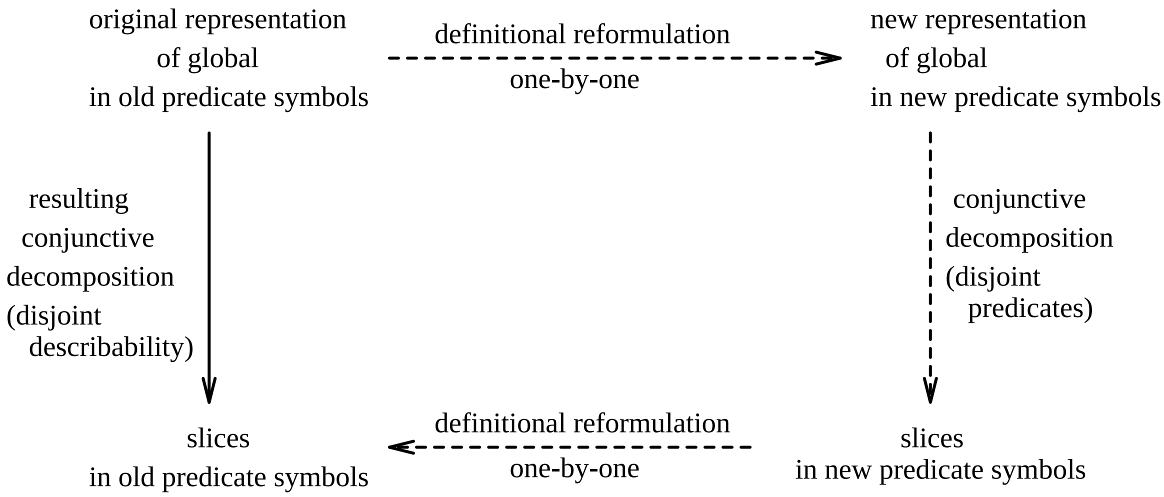

Figure 4: Disjoint Describability: a flow diagram of the reformulation steps involved.

7 Disjoint Describability and Disjoint Individuals

Next, we show how to use definitional reformulation to generalize the disjoint predicate special case: to the more general case of disjoint describability. More precisely, we use definitional reformulation to transform a disjointly describable global axiom set into a representation that has disjoint predicates, and then to transform back again after decomposition. Figure 4 illustrates. We show that the disjoint describability case, like disjoint predicate case, has a clean, partitioning conjunctive decomposition, which, moreover, implies interesting localities of inference and safeties of updating. We then identify an interesting special case of disjoint describability (asocial-monadic) that, like the disjoint predicate case, is easily recognizable in terms of the syntax of the starting global axiom set.

We begin with some preliminaries.

Definition 9 (Syndicate Reformulator)

We define a *syndicate* reformulator as a tuple of group reformulators that obeys an extra *syndication property*: their conjunction is also conservative.

More precisely: Let $ETT1[W, Y]$ and $ETT2[W, Z]$ each be a similar *tuple of tuples* of formulas; these formulas may be open or closed. Each element of the top level of tupling is itself a tuple of formulas cf. Definition 8. The top level tuple is thus a *syndicate* whose elements are groups of formulas.

Let $UT[W, Y, Z]$ be a tuple of closed formulas, of the same length as the top level tuples above. I.e., let it consist of one formula per group. Let $G[W]$ be a closed (background) formula, as in Definition 8.

Below, we use i to subscript groups, and j to subscript formulas within groups.

We say that $UT[W, Y, Z]$ is a syndicate reformulator between $ETT1[W, Y]$ and $ETT2[W, Z]$, given the background $G[W]$ when:

1. For each group i, UTi is a group reformulator between $ETTi$ and $ETT2i$ (given the background):
   $\forall i.\ UTi[W, Y, Z] \wedge G[W] \models$
   $\forall j.\ ETT1ij[W, Y] = ETT2ij[W, Z]$

2. the conjunction $UC \stackrel{\text{def}}{\equiv} \bigwedge_i UTi$ is conservative (given the background) with respect to Y and also with respect to Z:
   $G[W] \models \forall Y.\ \exists Z.\ UC[W, Y, Z]$
   $G[W] \models \forall Z.\ \exists Y.\ UC[W, Y, Z]$

The reason we call the above a *syndicate* reformulation is the linkage between the different groups imposed by the conjunction's (UC's) conservative extension property. This implies, but is not implied by, the conjunction of the conservative extension properties for each group's reformulator UTi.

Definition 10

(Partitioning Syndicate Reformulator)
We say that a syndicate reformulator cf. Definition 9 is *Z-partitioning* when:
$\forall i.\ UTi[W, Y, Z] \stackrel{\text{def}}{\equiv} Ui[W, Y, Zi]$
$\forall i, j.\ ETT2ij[W, Z] \stackrel{\text{def}}{\equiv} ETT2ij[W, Zi]$
where $\forall j \neq k.\ Zj \cap Zk = \emptyset$, i.e., the appearances of the symbols Z are partitioned by group.

Definition 11 (Disjoint Describability)

Suppose that UT is a Z-partitioning syndicate reformulator as in Definition 10, where for each group i, $ETT1i$ is defined as the concatenation of a (closed, base) formula $B1i$ with a tuple of (open, default) formulas $D1i$, and similarly, $ETT2i$ is the concatenation of $B2i$ and $D2i$.

Let $B1$ stand for the conjunction of the $B1i$'s; and $D1$ stand for the concatenation of the $D1i$'s. Let $B2$ and $D2$ be defined similarly.

Suppose also that $PDC(B2; D2; R; fix\ W; W, Z)$ fulfills the conditions in Theorem 1 (disjoint predicates), where the grouping, and the partition there on Z, is the same as in UT.

Then we say that $PDC(B1; D1; R; fix\ W; W, Y)$ is *disjointly describable* under (definitional) reformulation by $UT[W, Y, Z]$, given $G[W]$.

Theorem 12

(Clean Decomposition, given Disjoint Describability)
If a PDC is disjointly describable, then it is cleanly conjunctively decomposable into slices corresponding to the partitioning grouping employed in the reformulation. I.e., then the grouping employed in the reformulation forms the basis for a clean slicing.

More precisely: Suppose $PDC(B1; D1; R; fix\ W; W, Y)$ is *disjointly describable* under (definitional) reformulation by $UT[W, Y, Z]$, given $G[W]$, as in Definition 11. Then

$$PDC(B1; D1; R; fix\ W; W, Y) \equiv \bigwedge_i PDC(B1i; D1i; Ri; fix\ W; W, Zi)$$

where $Ri \stackrel{\text{def}}{=} R^{Ni}$ is the internal prioritization of the group of defaults $D1i$, whose index set (tuple) is Ni. (Equivalently, the Zi on the right hand side could be replaced by Z.)

Proof Overview: Theorem 1 plus some lemmas about definitional reformulation of circumscriptions. Figure 4 illustrates the logical flow of the proof. □

Theorem 12 immediately yields results about **locality of inference**, using Theorem 2, and about **safety** of updating, using Theorem 3.

Next, we consider a special case of disjoint describability: asocial-monadic.

Theorem 13

(Fixed Cases Reformulation of Defaults)
In PDC, defaults can be reformulated by relativizing them to fixed (-formula) cases.

More precisely: In a $PDC(B; D^N; R; fix\, W; Z)$, suppose that

$\forall i \in N.\ B[Z] \ \models\ \forall xi.\ \bigvee_{j=1}^{mi} Fij[Z, xi]$

where xi is a (possibly empty) tuple of individual (object) variables, and where, for each i, j, the (possibly) open (elementary) formula $Fij[Z, xi]$ is fixed relative to the circumscription (e.g., it mentions only function symbols; remember all functions are fixed). For each default index i, we call each Fij a *fixed case*. Suppose also that

$B[Z] \ \models$

$\forall i, j.\ \forall xi.\ Eij[Z, xi] \equiv (Fij[Z, xi] \supset Di[Z, xi])$

I.e., suppose that each Eij is equivalent to the default Di relativized to the fixed case Fij. Then

$PDC(B; D; R; fix\, W; Z) \ \equiv$

$PDC(B; E; RR; fix\, W; Z)$

where the tuple E stands for the concatenation of all the Eij's, and where RR is defined as the composition of R (as external prioritization) with a tuple $\emptyset T$ of empty prioritization p.o.'s. Each of $\emptyset T$'s elements is an empty prioritization p.o. $\emptyset Ti$ that is of size mi and corresponds to (i.e., has as domain) the index set of the (sub-) tuple Ei.

Proof Overview: The key is that each original default pre-order is equivalently reformulated, in the context of the circumscription's "augmentation" (i.e., second-order-quantified part in its definition cf. section 3), into a parallel default pre-order corresponding to Ei. □

Definition 14 (Asocially Monadic)

We say that a prioritized default circumscription $PDC(B; D; R; fix\, W; Z)$, or a corresponding CLD axiom set, is *asocially monadic* when:

1. All predicates in Z are monadic, i.e., 1-ary (a.k.a., unary).
2. The base sentence B has the form of a conjunction of universal[15] formulas. We will refer to these as the base formulas (axioms).
3. Every default formula (axiom) in D is quantifier-free.

[15]**Terminology:** By *universal*, we mean without existential quantifiers.

4. No base sentence (axiom) in B, and no default formula (axiom) in D, "mixes" individuals. I.e., in their clausal forms, no clause contains two literals with different arguments. **Intuition:** different individuals "don't want to have anything to do with each other", i.e., they are *"asocial"*.

5. All terms appearing in the base and default formulas are ground, except for primitive variables.

6. The prioritization R is either layered (e.g., parallel), or it is *point-modular* (see definition below).

7. All (explicit) fixtures are of predicate symbols (W), rather than of arbitrary formulas. (In addition, as usual, all function symbols are fixed.)

8. Uniqueness of Names Axioms (UNA): The base B includes axioms enforcing the distinctness of all terms that appear in the base and default axioms.

9. Besides in the UNA, equality does not appear in the base or default formulas. (Remember, equality, when viewed as a predicate, is binary, not monadic.)

Definition 15

(Point-Modular Prioritization)
Point-modular prioritization generalizes (i.e., the class includes) the prioritization that is typical in default inheritance networks. By "point" here, we mean an individual in the logical language, either named (a ground term, e.g. Ed) or unnamed (e.g., referred to by a first-order variable, e.g., x in $bat(x) \supset 2legs(x)$). (This idea of a point can be straightforwardly generalized to a *tuple* of individuals (e.g., $\langle Boss(4321), d \rangle$) to handle predicates / formulas with arity more than one; but we are only considering here the unary case in the context of the asocially monadic case.) By point-modular, we mean that the overall prioritization is equivalent to the composition of some external prioritization (over the points) composed with a tuple of internal prioritizations, one per point. Point-modularity results when the prioritization is only specified between the *same* instantiations of different defaults. E.g., when the bat default has higher priority than the mammal default at each point: (the default axiom whose default formula is) $bat(Betsy) \supset 2legs(Betsy)$ takes precedence over (the default axiom whose default formula is) $mammal(Betsy) \supset 4legs(Betsy)$,

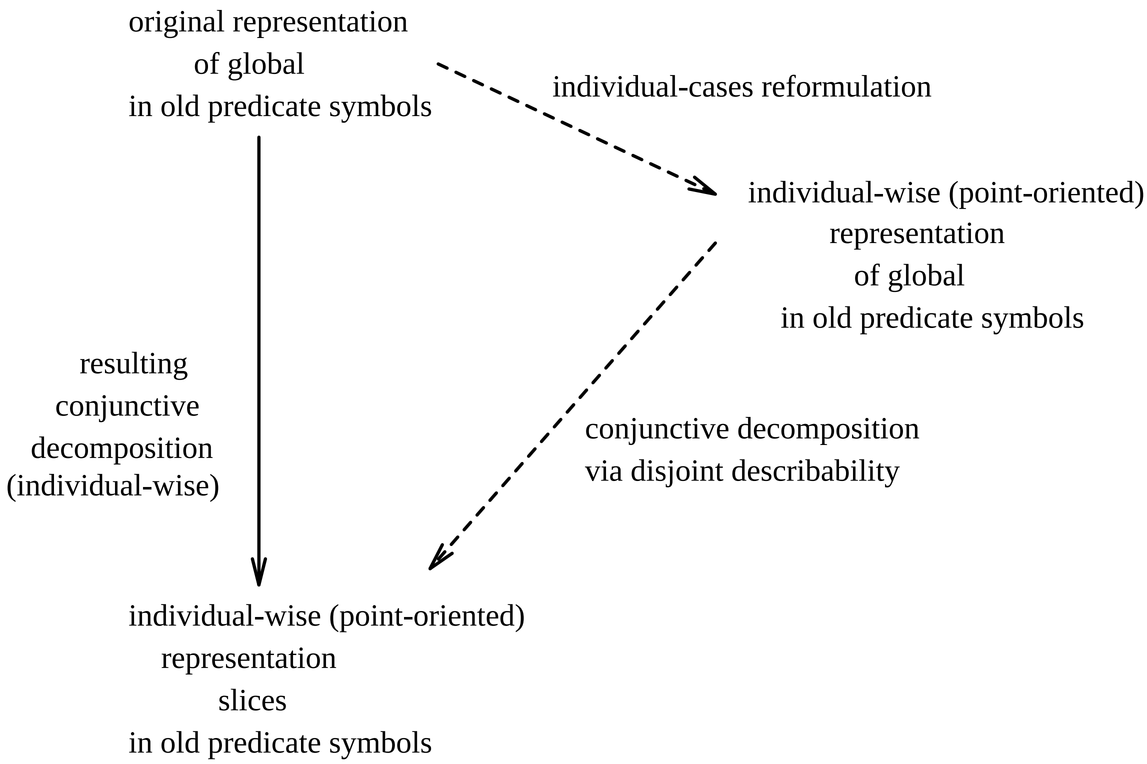

Figure 5: Asocially Monadic: a flow diagram of the reformulation steps involved. See also Figure 4.

$bat(Joe) \supset 2legs(Joe)$ takes precedence over $mammal(Joe) \supset 4legs(Joe)$, $bat(Fido) \supset 2legs(Fido)$ takes precedence over $mammal(Betsy) \supset 4legs(Betsy)$, etc., but there is no precedence between the defaults at different points, e.g., between $bat(Betsy) \supset 2legs(Betsy)$ and $mammal(Joe) \supset 4legs(Joe)$. Unfortunately, we do not have space to define point-modularity in further detail here; it requires discussing "pointwise" prioritization somewhat similar to that in [Lifschitz, 1988], and generalizing CLD to increase its expressivity with respect to prioritization. Note, however, that many point-modular prioritizations can be expressed in CLD. See [Grosof, 1992a] for more.

Theorem 16

(Decomposition by Reformulation, Individual-Wise)
Suppose the $PDC(B0; D0; R0; fix\ W; Z)$ is asocially monadic cf. Definition 14. Then the circumscription can be cleanly sliced, i.e., conjunctively

decomposed, into its *individual-wise* reformulation:

$$PDC(B0; D0; R0; fix\, W; Z) \;\equiv$$
$$\bigwedge_{j=1}^{m+1} PDC(B1j; D1j; Rj; fix\, W; Z)$$

This individual-wise reformulation is **defined** as follows.

The basic idea of the reformulation is to divide the base and default axioms into groups: one group per named individual, plus a catch-all "remainder" group for all other, unnamed individuals. Some reformulation, of a relatively simple kind that is different from decomposition and one-by-one definitional reformulation, is involved in order to break up the quantified base axioms and the open defaults into these cases. Figure 5 illustrates this logical flow. The details of the overall reformulation are, however, a bit involved to define; bear with us.

To begin with, we partition the base and default formulas according to which arguments appear in them.

Let $J \stackrel{\text{def}}{=} \{1, \ldots, m\}$ index the set of all ground terms aj that appear in the base or default formulas.

Let $B0j$ stand for the tuple of base formulas that mention aj. Each of its members we write as $B0jk[Z]$.

Let $B0V$ stand for the tuple of base formulas, other than the UNA, that mention a free variable (all of these are universally quantified). Each of its members we write as $\forall x.\, B0Vk[Z, x]$. Here x is a single (free) individual variable.

We treat the default formulas similarly to the base. Let $D0j$ stand for the tuple of default formulas that mention aj. Each of its members we write as $D0jk[Z]$.

Let $D0V$ stand for the tuple of default formulas that mention a free variable (i.e., that are open; all of these are quantifier-free). Each of its members we write as $D0Vk[Z, x]$. Here x is a single (free) individual variable.

Next, we reformulate the base and default formulas that mention a variable.

For each $j \in J$, let $B1Vj$ stand for the instantiation of the quantified base formulas $B0V$ to aj. Each of its members $B1Vjk[Z]$ is defined as the formula $B0Vk[Z, aj]$.

Let $UNNAMED[x]$ stand for the formula
$\bigwedge_{j \in J} x \neq aj$.

Let $B2V$ stand for the tuple of quantified base formulas after relativization to the unnamed case. Each of its members $B2Vk[Z]$ is defined as:

$$\forall x.\, UNNAMED[x] \supset B0Vk[Z, x]$$

For each $j \in J$, let $D1Vj$ stand for the instantiation of open default formulas $D0V$ to aj. Each of its members $D1Vjk[Z]$ is defined as the formula $D0Vk[Z, aj]$.

Let $D2V$ stand for the tuple of open default formulas after relativization to the unnamed case. Each of its members $D2Vk[Z]$ is defined as:

$UNNAMED[x] \supset D0Vk[Z, x]$

For each $j \in J$, Let $B1j$ stand for the conjunction of (all members of) the tuples $B0j$ and $B1Vj$.

For $j = m+1$ (i.e., the unnamed case), let $B1m+1$ stand for the conjunction of (all members of) the tuple $B2V$ plus the UNA.

For each $j \in J$, Let $D1j$ stand for the concatenation of the tuples $D0j$ and $D1Vj$.

For $j = m+1$ (i.e., the unnamed case), let $D1m+1$ stand for the tuple $D2V$.

Let Rj be defined as the prioritization internal to $D1j$, i.e., as R^{Nj}, where, for each $j = 1, \ldots, m+1$, Nj is the index tuple of $D1j$.

Proof Overview: We use a first stage of reformulation employing Theorem 13. This stage involves what we called above an "extra" kind of reformulation: e.g., to reformulate each open default axiom and each quantified base axiom into a collection of "point"-case (individual-case) axioms, plus a remainder-case (unnamed case) axiom. Then we use a second stage partitioning syndicate reformulation into disjoint describability, employing Theorem 12. In that second stage of reformulation, we treat the UNA as background. There, the newly introduced predicates are all 0-ary, except for those corresponding to the catch-all case. **The definitional reformulator consists of the explicit definitions of these newly introduced predicates. There is one new predicate for each ground atom in the original representation.** Note that the second stage itself combines two kinds of reformulation: definitional reformulation, to transform into a representation with disjoint predicates, and conjunctive decomposition. □

Figure 5 illustrates the logical flow of the reformulation steps involved; it builds upon Figure 4.

Application to Main Example: (Continued from the discussion in section 5:) Consider our main motivating example (about legged-ness and meetings, from section 2). There, after the final update $\mathcal{U}_8$ (and, indeed, at any earlier point in the sequence of updates), the legged-ness slice, i.e., the set of axioms about legged-ness ($\mathcal{U}_1$, $\mathcal{U}_2$, and $\mathcal{U}_4$) is asocially monadic. It can thus

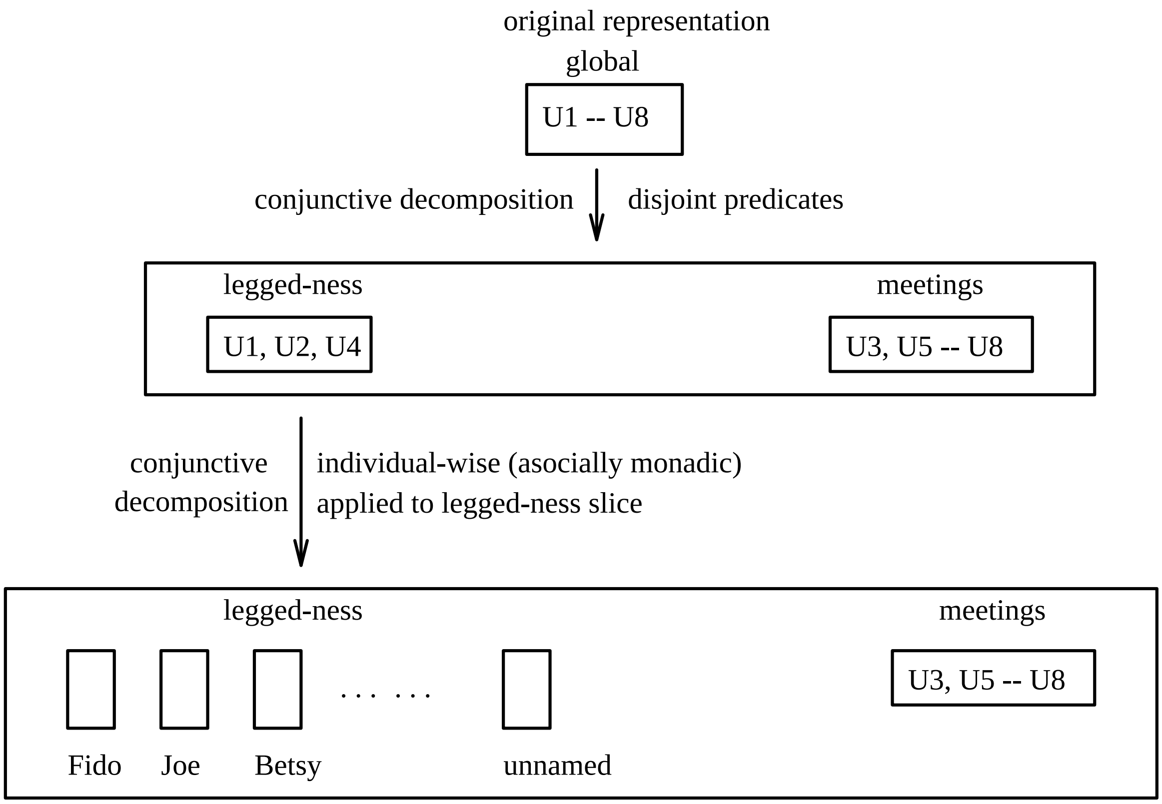

Figure 6: Conjunctive Decomposition using Asocially Monadic and Disjoint Predicates: Each outer box stands for a decomposition. Each inner box stands for a constituent axiom set.

be conjunctively decomposed cleanly, individual-wise. Figure 6 illustrates this decomposition, which we explain next.

In our main motivating example (section 2), we can conjunctively decompose the global axiom set (after the last update $\mathcal{U}_8$) into two slices by employing the disjoint predicates result (Theorem 1): one slice about legged-ness, and the other slice about meetings. This first-stage decomposition is the same as in Figure 3. We can conjunctively decompose the legged-ness slice, individual-wise, by employing the asocially monadic result (Theorem 16). That is, in a second stage, we slice (more finely) within a slice that arose from the first stage. The second stage thus yields a second, finer-grain decomposition of the global axiom set, containing the meetings slice (unchanged from the first stage) plus each of the individual-case legged-ness slices. Together, the two stages exemplify the ability to decompose hierarchically / recursively. Each of the named-individual / "point" slices in the second stage contains a set of axioms that correspond to the instantiation / particularization of the original legged-ness axioms ($\mathcal{U}_1$, $\mathcal{U}_2$, and $\mathcal{U}_4$) to (the case of) one named individual, e.g., *Joe*.

As we discussed earlier, the definitional reformulation involved in the individual-wise decomposition cf. Theorem 16 introduces a new 0-ary predicate for each ground atom in the original representation; in this example, two such new predicates are:

$$nbatJoe \equiv bat(Joe)$$
$$n2legsJoe \equiv 2legs(Joe)$$

Theorem 17

(Individual-wise Locality of Inference, when Asocially Monadic)
In Theorem 16, each slice j, where j is the (index of) a named individual (cf. statement of that Theorem), is sound and complete, relative to the global theory, for inference over its corresponding sub-language. That sub-language consists of the ground formulas (sentences) in which the only individual mentioned is j (e.g., *Betsy*). This locality holds both for forward inference, and for backward inference (query-answering). Note that to perform inference using any subset SJ of the named individuals J, one need only work in the conjunctive combination of those slices corresponding to SJ.

For query-answering about a new named individual b (named in the query), just introduce the new term b into the set of terms that are indexed by J in the theorem. The only additional requirement is that the UNA ensure its distinctness from the other named individuals.

Application to Main Example: Thus after each update, inferences about any named individual's (e.g., *Joe*'s) legged-ness can be made by working in a slice axiom set that has been instantiated / particularized to that individual (*Joe*). One advantage of this is that simpler inference algorithms are available for such an expressively simpler axiom set. In this case, there is a decidable polynomial-time procedure (see "total-propositional" case results in [Grosof, 1992b]). By contrast, there is no general inference procedure, even for query-answering, yet available for the full example (i.e., including the meetings aspect). (See next section for discussion of inference procedures available for prioritized circumscription.) This illustrates that decomposition-type reformulation is useful to exploit available / known tractable special cases to do part of the inference in a NM theory, even when the overall theory is intractable or undecidable (see next section for more discussion of this point.)

Theorem 16 also immediately yields a powerful result about belief revision.

Theorem 18

(Safety of Updating, when Asocially Monadic)
In CLD, let the previous axiom set be asocially monadic. Let an update $\mathcal{U}$ consist of base, default, and prioritization axioms, such that the formula parts of the base and default axioms are ground and mention only a set of named individuals IU. Then all of the previous conclusions derived solely from the rest of the named individuals' slices (i.e., the slices according to Theorem 16) are safe under the update.

Application to Main Example:
E.g., after update $\mathcal{U}_4$ (mentioning only Joe and $Spot$), this theorem tells us that we do not have to re-consider whether the previous conclusion $2legs(Betsy)$ is still sanctioned: it must be preserved. Thus we can know, with relatively little computational work (see discussion of complexity in next section), that most of the previous NM conclusions are safe.

Disjoint Groups of Individuals:
Definition 14 and Theorems 16, 17, and 18 also generalize straightforwardly to considering disjoint *groups* of individuals, where any syntactic mixing in the axioms involves only individuals within the same group.

8 Discussion, Conclusions, and Future Work

Proof Procedures: Prioritized default circumscription is expressively reducible to prioritized predicate circumscription (see section 3). There exist several backward proof procedures for fairly expressive classes of prioritized predicate circumscription, including for layered (stratified) prioritization [Przymusinski, 1989] [Ginsberg, 1989] [Baker and Ginsberg, 1989] [Inoue and Helft, 1990] [Inoue *et al.*, 1991]. More interestingly, [Geffner, 1989] contains a proof theory and proof procedures which promise to be easily adaptable (using an equivalence theorem reported in [Grosof, 1991], detailed in [Grosof, 1992b]) to circumscription with *non*-layered prioritization.

Related Work: Note that we emphasize updating with new defaults, not just new for-sure axioms, unlike the conditional approaches to NMR (e.g., [Kraus *et al.*, 1990]). The ideas and results here apply to other NM formalisms, e.g.: Default Logic and Poole's [1988] and Brewka's [1989] systems, via the equivalence result in [Lifschitz, 1990]; as well as Geffner's [1989] system. The closest idea to conjunctive decomposition in the previous NMR literature is [Rathmann, 1990], who focussed, however, on conjunctively integrating heterogeneously-specified circumscriptive theories. He considered, moreover, only layered-priority predicate circumscriptions. Rathmann's and our work was developed independently. We are unaware of any other applications of reformulation to non-monotonic reasoning.

More Decompositions and Safeties: We did not have space here to report a number of additional results [Grosof, 1992b] about decompositions and their implications for safeties of updating, including about higher prioritization, hypotheticals, syntactic positivity, "serial" decompositions, weaker forms of irrelevance; and about the relationship of decompositions to specification and backward inference.

Algorithms and Automation of Our Results: In future work, we plan to automate *recognition* of decompositions and safeties of updating cf. our theorems, and the actual performance of the according *reformulation*. For the disjoint-predicates and asocial-monadic cases, we have **polynomial-time algorithms** to perform this: $O(n^3)$ time, where n is the size of the CLD axiom set.

Exploiting Truth Maintenance: Such recognition establishes "monotonicity" (i.e., implication) relationships between theories and sub-theories (e.g., theory after update versus theory before update; or theory versus slice). We plan also to automate a generalized ATMS-style [de Kleer, 1986] high-level architectural book-keeping scheme to exploit such stored monotonicity relationships to support inference and belief revision in a prioritized database. [Grosof, 1992b] gives details.

More General Cases of Disjoint Describability: In future work, we aim to find cases of disjoint describability that are more general than asocial-monadic, but are still easily recognizable syntactically (in terms of the syntax of the global axiom set). E.g., in our main example, it would be nice to be able to particularize the Meetings slice to the individual Ed, in the same way that the asocial monadic result guarantees one can particularize the Leggedness slice to the individual Joe. Right now, we can show this particularization about Ed is legitimate, but our proof method is by hand. We would like to be able to formalize and automate a class of decompositions for which this (Ed etc.) is an instance.

Conclusions I: See **Strategy and Summary** in section 1.

Conclusions II: Analyzing Computational Advantages of Reformulation: In future work, we also plan to analyze in detail the computational advantages and trade-offs involved in our decompositions and definitional reformulations. You may be wondering why we did not give any such computational complexity analysis in this paper. The main reason is that the picture is quite complicated for non-monotonic reasoning.

Even for query-answering in propositional default theories without priorities, current results show worst-case is **exponential** (NP-hard) [Selman and Kautz, 1989] [Kautz and Selman, 1989] [Selman and Levesque, 1989]. Thus: *Divide-to-conquer, i.e., seeking locality, is clearly desirable.*

But the basic complexity results for any kind of forward reasoning with priorities, for any kind of belief revision, and even for most kinds of backward (query-answering) reasoning are not available for circumscription, or other NM formalisms. Known tractable cases are highly restricted. ([Selman and Kautz, 1989] [Kautz and Selman, 1989] give polynomial-time backward procedures for special cases, including restrictions of Horn, of propositional default reasoning in their model-preference logic and in Default Logic. Delgrande [1991] gives a polynomial-time backward procedure for a Horn propositional case of his conditional logic.)

However, we believe that as these results become available, we will be able to show that decomposition and reformulation are advantageous. Our aim has been to develop methods that will be broadly applicable, and to break off a piece of the overall hard problems of non-monotonic reasoning. In current work, we are addressing how to relate our results to currently known tractable and intractable cases.

One clear advantage is that for many cases with quantification, for which worst-case is **undecidable** ([Reiter, 1980] [Kolaitis and Papadimitriou, 1988]): *We are able to reformulate some of the reasoning to become propositional, hence decidable.* E.g., when reasoning about individuals, for the asocially monadic class of theories (see Theorem 16).

Acknowledgements

Thanks to Devika Subramanian, Vladimir Lifschitz, and Michael Lowry for long-ago useful discussions on logical aspects of definitional reformulation. Thanks to Leora Morgenstern, Hector Geffner, and two anonymous reviewers for their comments on previous drafts.

References

[Baker and Ginsberg, 1989] A. Baker and M. Ginsberg. A theorem prover for prioritized circumscription. *Proceedings IJCAI-89*, pages 463–467, Detroit, MI., 1989.

[Brewka, 1989] G. Brewka. Preferred subtheories: An extended logical framework for default reasoning. *Proceedings IJCAI-89*, pages 1043–1049, Detroit, Michigan, 1989.

[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.

[de Kleer, 1986] J. de Kleer. An assumption-based truth maintenance system. *Artificial Intelligence*, 28:280–297, 1986.

[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.

[Doyle, 1979] J. Doyle. A truth maintenance system. *Artificial Intelligence*, 12:231–272, 1979.

[Geffner, 1989] H. Geffner. *Default Reasoning: Causal and Conditional Theories.* PhD thesis, Computer Science Department, UCLA, Los Angeles, CA, 1989. Revised version published by MIT Press, 1992.

[Ginsberg, 1989] M. Ginsberg. A circumscriptive theorem prover. *Artificial Intelligence*, 39:209–230, 1989.

[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. Generalizing prioritization ii (working title). Working paper., 1992.

[Grosof, 1992b] Benjamin N. Grosof. *Updating and Structure in Non-Monotonic Theories*. PhD thesis, Computer Science Dept., Stanford University, Stanford, California 94305, 1992.

[Inoue and Helft, 1990] Katsumi Inoue and Nicolas Helft. On theorem provers for circumscription. Working paper. This is a revised version of a paper appearing under the same title in the Proceedings of the Canadian Conference on Computer Science and Artificial Intelligence '90, Ottawa, Canada, May 1990., Apr 1990.

[Inoue *et al.*, 1991] Katsumi Inoue, Nicolas Helft, and David Poole. Query answering in circumscription. *Proceedings of IJCAI-91*, pages 426–431, San Mateo, California, 1991. Morgan Kaufmann.

[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.

[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.

[Kraus *et al.*, 1990] S. Kraus, D. Lehmann, and M. Magidor. Preferential models and cumulative logics. *Artificial Intelligence*, 44:167–207, 1990.

[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, 1987] Vladimir Lifschitz. On the declarative semantics of logic programs with negation. In Matthew L. Ginsberg, editor, *Readings in Non-monotonic Reasoning*. Morgan Kaufmann, San Mateo, CA, 1987.

[Lifschitz, 1988] V. Lifschitz. Circumscriptive theories: a logic-based framework for knowledge representation. *Journal of Philosophical Logic*, 17:391–441, 1988.

[Lifschitz, 1990] Vladimir Lifschitz. On open defaults. *Proceedings Symposium on Computational Logic*, Brussels, Belgium, 1990.

[McCarthy, 1986] J. McCarthy. Applications of circumscription to formalizing commonsense knowledge. *Artificial Intelligence*, 28:89–116, 1986.

[Moore, 1985] R. Moore. Semantical considerations on non-monotonic logics. *Artificial Intelligence*, 25:75–94, 1985.

[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.

[Rathmann, 1990] Peter K. Rathmann. *Nonmonotonic Semantics for Partitioned Knowledge Bases*. PhD thesis, Computer Science Dept., Stanford University, Stanford, California 94305, Jun 1990.

[Reiter, 1980] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 12:81–132, 1980.

[Selman and Kautz, 1989] 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 Levesque, 1989] B. Selman and H. Levesque. The tractability of path-based inheritance. *Proceedings IJCAI-89*, pages 1140–1145, Detroit, MI., 1989.

[Touretzky, 1986] D. Touretzky. *The Mathematics of Inheritance Systems.* Pitman, London, 1986.