Contents

To appear at: the Second International Workshop on
Logic Programming and Non-Monotonic Reasoning.
Held at Gulbenkian Foundation, Lisbon, Portugal, June 28–30, 1993.
November, 1992; revised June 25, 1993
Soon to be available as IBM Research Report

Sympathetically Solitary Default Theories: A New Case of "Easy" Non-Monotonic Reasoning — Preliminary Report

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

Abstract

Expressively rich non-monotonic reasoning is significantly more computationally difficult, in the worst case, than monotonic reasoning (over the same base language). An interesting direction of research, therefore, is to find expressively restricted cases for which it is relatively easy. In this paper[1], we give a new such case: the class of what we call sympathetically solitary default theories. For this class, we are able to give decidable and tractable methods for the non-monotonic aspect of reasoning. The sympathetically solitary class appears useful for representing reasoning that is shallow in a particular sense. Its tractability (in the non-monotonic aspect) arises from the lack of conflict between multiple defaults, similarly to stratified logic programs with negation by failure (there the minimization of each predicate or ground atom corresponds to a default).

We develop the definition and results about the sympathetically solitary class in terms of circumscription; however, they also apply straightforwardly to Default Logic, Autoepistemic Logic, and several other non-monotonic formalisms as well.

The sympathetically solitary class generalizes Lifschitz' solitary class. It also includes, as a special case, the simple application of the Closed World Assumption to unit-clausal databases, when there is domain closure and a complete theory of equality, e.g., uniqueness of names. However, sympathetically solitary overlaps with, and neither includes nor is included by, the following classes: logic programs with negation by failure; Horn; default inheritance cf. Touretzky; predicate completion; and the general Closed World Assumption.

[1]adapted and revised from parts of [Grosof, 1992]

1 Introduction

Expressively rich non-monotonic reasoning is significantly more computationally difficult, in the worst case, than monotonic reasoning (over the same base language). An interesting direction of research, therefore, is to find expressively restricted cases for which it is relatively easy. In this paper[2], we give a new such case: the class of what we call sympathetically solitary default theories. For this class, we are able to give decidable and tractable methods for the non-monotonic aspect of reasoning. The sympathetically solitary class appears useful for representing reasoning that is shallow in a particular sense. Its tractability (in the non-monotonic aspect) arises from the lack of conflict between multiple defaults, similarly to stratified logic programs with negation by failure [Apt *et al.*, 1987] [Van Gelder, 1988] (there the minimization of each predicate or ground atom corresponds to a default).

We develop the definition and results about the sympathetically solitary class in terms of circumscription; however, they also apply straightforwardly to Default Logic [Reiter, 1980], Autoepistemic Logic [Moore, 1985], and several other non-monotonic formalisms as well.

The sympathetically solitary class generalizes Lifschitz' [1985a] solitary class. It also includes, as a special case, the simple application of the Closed World Assumption [Reiter, 1978] to unit-clausal databases, when there is domain closure and a complete theory of equality, e.g., uniqueness of names. However, sympathetically solitary overlaps with, and neither includes nor is included by, the following classes: logic programs with negation by failure; Horn; default inheritance cf. Touretzky [1986]; predicate completion [Clark, 1978]; and the general Closed World Assumption.

In the next 3 sections, we cover some preliminaries and give background.

2 Prioritized Default Circumscription: Definitions and Notation

Circumscription in its predicate form was defined by John McCarthy [1980] [1986]. McCarthy [1986] proposed the concept of prioritization in circumscription, which Vladimir Lifschitz [1985a] first formally defined, and which we generalized in [Grosof, 1991] and studied in [Grosof, 1992]. Circumscription in its general form was defined by Lifschitz [1984] as minimization with respect to a (preference) pre-order. A **pre-order** is a transitive and reflexive binary relation. In [Grosof and Russell, 1990] [Grosof, 1992] we defined, and in [Grosof, 1992] we studied, *prioritized default* circumscription. By a default here, we mean a formula that is maximized (see details below). Prioritized default circumscription is a slight, but convenient, generalization of prioritized predicate circumscription cf. [Grosof, 1991], which generalized the previous concept of prioritization in predicate circumscription that was defined by Lifschitz [1985a] and inspired by McCarthy [1986].

Prioritized default circumscription (PDC) is a fairly expressively rich non-monotonic logical formalism, useful for representing a broad variety of applications, including in machine learning; see [Grosof and Russell, 1990] and [Grosof, 1992] for discussion of these. Sympathetically-solitary is an interesting expressive restriction within the more expressive

[2] adapted and revised from parts of [Grosof, 1992]

class of full PDC. Next, we review PDC.

Notation: Let $G \leq E$ stand for $\forall x.\ G(x) \supset E(x)$, where G and E are open (first-order) formulas with the same arity (i.e., are **similar**), and x stands for a tuple of individual object variables.

Let $G{=}E$ stand for $(G \leq E) \wedge (E \leq G)$.

Let $G \leq E$ stand for $(G \leq E) \wedge \neg(E \leq G)$.

We also apply this notation to tuples $G = \langle G_1, \ldots, G_m \rangle$ and $E = \langle E_1, \ldots, E_m \rangle$, assuming that G and E are similar: $G \ \leq \ E$ stands for

$$G_1 \leq E_1 \wedge \ \ldots \ \wedge G_m \leq E_m \ .$$

Let $\mathcal{L}$ be a first-order language; we will call it "the base language". Let Z be the tuple of all predicate and function symbols in $\mathcal{L}$. Let $B[Z]$ be a closed sentence of $\mathcal{L}$, representing a conjunctive collection of some for-sure premise beliefs. (Notationally, we employ square brackets to explicitly indicate which symbols appear in a formula.) Let $D[Z]$ be a tuple of formulas (open or closed) of $\mathcal{L}$, representing a collection of some default premise beliefs. Let the tuple N index the tuple D. Let R be a strict partial order of prioritization, defined over the domain N, where $R(j, i)$ means that j has strictly higher priority than i; i.e., the maximization of the default formula Dj has strictly higher prioritization than the maximization of the default formula Di. Let $F[Z]$ be a tuple of formulas (open or closed) of $\mathcal{L}$, representing a collection of some "fixtures" (see discussion of fixing below). (Z, B, D, R, and F are all finite.)

For any tuple Y of symbols, we say Y is **similar** to Z when Y has the same number of symbols as Z and each symbol in Y has the same type (predicate versus function) and arity as its correspondent in Z. For any wff E in $\mathcal{L}$, $E[Y]$ stands for E with the symbols Y substituted for Z (where Y is similar to Z).

We **define** the prioritized default circumscription (**PDC**)

$PDC(B; D; R; fix\ F; Z)$

as standing for its **circumscription formula**:

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

This formula is second-order. ($B[Z']$ stands for $B[Z]$ with Z' substituted for Z, and Z' is quantified over.) Here, $\prec_{(D;R)}$ means strictly less preferred with respect to the following prioritized default pre-order:

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

Here Dj and Di refer to the j^{th} and i^{th} members, respectively, of the tuple D. $\preceq_{(D;R)}$ is a *non*-strict preference relation; its strict version is:

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

In the circumscription formula above, the right-hand-side second-order-quantified expression is called the **augmentation part** of the circumscription. B is called the **base** of the circumscription. D are called the **defaults** of the circumscription. R is called the **prioritization (partial order)** of the circumscription. F are called the **fixtures** of the circumscription.

Circumscriptive Language of Defaults (CLD): In [Grosof and Russell, 1990] and [Grosof, 1992] we developed a convenient meta-language for specifying PDC's: the Circumscriptive Language of Defaults (CLD). In CLD, there are four kinds of axioms, i.e., premises. 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 $(d2)$. A $\Box\!>$ prefix indicates that the formula that follows is a *fixture* axiom.

A set of CLD axioms, i.e., a CLD **axiom set**, specifies a PDC of the form given above: B is the conjunction of the sentences specified in the base axioms; D is the tuple of the formulas specified in the default axioms; F is the tuple of the formulas specified in the fixture axioms; R is the transitive closure of the acyclic binary relation whose pairs are specified in the prioritization axioms.

Models and Theories: The essence of the definition of general circumscription is that the *pre-order represents a preference criterion*. For prioritized default circumscriptions, the overall preference pre-order (let us call it $\preceq_H$), defined on tuples similar to Z, corresponds to a preference pre-order $\sqsubseteq_H$ defined on *models* (e.g., M). By "equivalent correspondence" here, we mean that the models of the circumscription are exactly those models, of those satisfying the base sentence $B[Z]$, that are most preferred with respect to $\sqsubseteq_H$. In reading such a circumscription, one can thus view a symbol tuple Z as a syntactic "stand-in" for a model M.

We **define** the circumscriptive prioritized default **theory** corresponding to the PDC above as the set of all **conclusions** entailed (model-theoretically, in second-order logic) by the PDC.

Fixing: "Fixing" is an expressively important technical aspect in the definition and application of various historical variants of circumscription, e.g., in [McCarthy, 1986] [Lifschitz, 1985a]. Most use of fixing has been to fix predicate and/or function symbols; our definition of PDC above is a bit more general in that it permits the fixing of any first-order formula. In this paper, we mainly do not use this generality, however.

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 circumscription. 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. In most previous applications of circumscription, all function symbols have been fixed. 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.

In terms of our definition above of PDC, fixing of predicate and/or function symbols can be specified via the fixture formulas (F). For any predicate symbol P, fixing the formula $P(x)$ has the equivalent effect of fixing that symbol. For any function symbol f, fixing the formula $y = f(x)$ has the equivalent effect of fixing that symbol. (Here x is a tuple of object variables, of appropriate arity, and y is a single object variable.)

In this paper, we mainly will fix only symbols. Let $W \subseteq Z$ be a tuple of predicate and/or function symbols. As **shorthand for fixing symbols**, we simply write

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

to indicate the fixing of W.

In [Grosof, 1992] (Theorem 2.24 there), we show that domain closure plus a complete theory of equality about the ground terms, e.g., uniqueness of names, implies that all functions are effectively fixed. By "effectively", we mean that fixing the functions does not affect the circumscription, in the sense of equivalence.

See [Grosof, 1991] and [Grosof, 1992] for more discussion of how prioritized circumscriptions are defined. Note that the prioritization partial order R is not necessarily layered (stratified) as it was in [Lifschitz, 1985a].

Predicate Circumscription: Prioritized *predicate* circumscription is the special case of prioritized default circumscription in which each default formula Di takes the form of a negative open literal $\neg Pi(x)$, corresponding to the minimization of a predicate (symbol) Pi. We write it as follows:

$$PPC(B; P; R; fix\, W; Z) \stackrel{\text{def}}{\equiv} PDC(B; E; R; fix\, W; Z)$$

where $\forall i \in N.\ \ Ei \stackrel{\text{def}}{=} \neg Pi$ and P is the tuple of all the Pi's. Historically, in the context of defining predicate circumscription, McCarthy [1986] and Lifschitz [1985a] [1984] only considered fixing of symbols. Predicate symbols that are neither minimized nor fixed (i.e., the symbols $Z - P - W$ above) are called **auxiliary** predicate variables.

Why Employ PDC and CLD: We employ PDC and CLD to clarify the definitions of axiom sets and of updating, and the intuitive relationship to other formalisms for default reasoning.

Abnormality-Style PPC: [Grosof, 1992] (section 3.3) shows the essential equivalence of any prioritized *default* circumscription to a corresponding, "abnormality-style", prioritized *predicate* circumscription, generalizing a previous result (for the case without prioritization) that appeared in [Lifschitz, 1984]. Next, we repeat that result.

Definition 1 (Abnormality-Style; Abnormality Theory)

Let Y be a tuple of predicate and function symbols. Let ab be a tuple of predicate symbols that is distinct/disjoint from Y. 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 PPC is **abnormality-style** if it has the **adaptive form**

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

or the **definitional form**

$$PPC(B[Y] \wedge (\neg ab = E[Y]); ab; R; fix\, W; Y, ab)$$

where the fixed symbols W are a subset of Y. Here, $\neg ab$ stands for the tuple whose i^{th} member is $\neg abi$, where abi is the i^{th} member predicate of the tuple ab. We also say that the abnormality-style PPC is an **abnormality theory**.

Theorem 2 (Equivalence of Abnormality-Style PPC to PDC)
Suppose a PPC is abnormality-style cf. Definition 1. Then the PPC is equivalent to:

$$(\neg ab = E[Y]) \ \wedge \ PDC(B; E; R; fix\, W; Y)$$

3 Computational Complexity, Exhaustiveness

Known computational complexity results indicate that 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 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], and several other non-monotonic modal logics [Gottlob, 1992]. [3]

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 co-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**. We discovered the sympathetically solitary class after specifically setting out to look for a case where there is no conflicting interactions between defaults.

Previously 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].

[3] By *skeptical* entailment in Default Logic, we mean entailment by the intersection of all the (credulous) "extensions" of the Default Logic "default theory", i.e., truth in all models of all extensions. By skeptical entailment in other credulous formalisms, e.g., Autoepistemic Logic, we mean in a similar sense.

We **observe** that 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, relative to the expressively rich worst-case, *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 PDC is a *second*-order formula. However, often, a PDC 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:

$$PDC(B; D; R; fix\ F; Z) \equiv B[Z] \wedge GA[Z]$$

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 without priority, i.e., the Nixon diamond, a generating axiom for the circumscription is:

$$\begin{aligned}&[\forall x.\ Republican(x) \wedge (x \neq Nixon) \supset \neg Pacifist(x)]\\ &\quad \wedge\ [\forall x.\ Quaker(x) \wedge (x \neq Nixon) \supset Pacifist(x)]\end{aligned}$$

More generally, by "generating axiom**s**", 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 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[4].

Similar notions of reduction, exhaustiveness, and selectiveness can be defined for many other NM formalisms, including the skeptical versions of Default Logic and Autoepistemic Logic.

4 Relationships of PDC to Other Non-Monotonic Formalisms

There are many known relationships between prioritized default circumscription and many other non-monotonic logical formalisms: see, for example, the review in [Grosof, 1992]. For example, in [Grosof, 1992] (section 8.5), we give the following slight generalization of Etherington's [Etherington, 1988] Theorem 8.3 (which is for the predicate case).

Observation 3 (Overlap between Default Logic and Circumscription)

Suppose the for-sure beliefs B entail: 1) domain closure; and 2) a complete theory of equality about the ground terms, e.g., uniqueness of names.
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.
□

As Konolige [1988] and others since have shown, there is a close relationship between Autoepistemic Logic (AEL) 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.

The equivalence mapping to Default Logic, in the Observation above, thus also carries over to Autoepistemic Logic.

Other non-monotonic formalisms to which PDC has a close relationship include: model-preference logics [Shoham, 1988] [Brown and Shoham, 1989] [Boddy *et al.*, 1989] [Selman and

[4]see section 5.8 of [Grosof, 1992] for a precise notion of completeness with respect to a subset of defaults

Kautz, 1989b] (recall our earlier review of PDC's model theory), Poole's [1988] formalism (which is equivalent to a special case of Default Logic) and Ginsberg's [1988] Multi-Valued Logic (e.g., see [Ginsberg, 1989]).

5 The Solitary Class of Predicate Circumscription

The sympathetically solitary class is a generalization of Lifschitz' *solitary* class.

Definition 4 (Solitary (Lifschitz))

Lifschitz [1985a] 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 4 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 5 (Solitary (Lifschitz))

Lifschitz [1985a] 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.

The solitary case also has an interesting relationship to the **Closed World Assumption** (CWA) [Reiter, 1978]. A simple, basic case of the CWA is when all clauses are unit, and both domain closure and uniqueness of names are present. This case corresponds equivalently to a special case of solitary predicate circumscription, in which P is the tuple of all predicate symbols, and W is the tuple of all function symbols. This correspondence follows from Lifschitz' result [1985b] (Theorem 3.1 there) relating the CWA to predicate circumscription.

Observation 6 (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 Lifschitz [1987] showed is equivalent to a special case of PPC). 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.

2. **Explicit Solution and Collapse to First-Order:** Theorem 5 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 (section 3) 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. **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 section 7.

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 7 (Eliminating Auxiliaries (Lifschitz))

Lifschitz showed ([1985a] Proposition 2') that

$$PPC(B[P,Y,W]; P; R; fix\, W; P,Y,W) \equiv$$
$$B[P,Y,W] \wedge PPC((\exists Y'.\ B[P,Y',W]); P; R; fix\, W; P,W)$$

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 sub-language. While performing this existential projection is feasible by hand for small examples, we do not know of any practical computational method to automate it for large base axiom sets.[5] Therefore, we would like to find some other, practical method of exploiting the nice features of the solitary case.

Definition 8 (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 4 to permit additional (distinct) predicate variables Y to appear in NEG. 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 9 (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 in abnormality theories (Definition 1). 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 case of prioritized predicate circumscription [Lifschitz, 1987]. Like the solitary case, they are conflict-free. However, in stratified logic programs with negation, there are (typically) no fixed predicates. 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.

6 Sympathetically Solitary Class of Default Circumscription

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) 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

[5]Perhaps such a method can be developed, however, based on the approach of [Gabbay and Ohlbach, 1992].

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.

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.

Definition 10 (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 $Y \stackrel{\text{def}}{=} Z - Q - W$ are the varied symbols 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.

Rules-and-Facts Form: An interesting special case of sympathetically solitary is when: 1) the fixed symbols are exactly the functions (recall from section 2 that domain closure and uniqueness of names, for example, imply that all functions are fixed); 2) D and S take the form of clauses, i.e., "rules"; and 3) $(\sigma Q \leq G[W])$ takes the form of ground unit clauses, i.e., "facts". Our example, next, is in this form.

Example 11 (Socializing, Sympathetically Solitary)

Consider the following CLD axiom set $\mathcal{A}_1$ about Socializing. $\mathcal{A}_1$ contains two defaults: that pleasant-ness is returned by politeness, and that powerful people are treated politely. And $\mathcal{A}_1$ contains some assorted for-sure information, including information about the predicates mentioned in those defaults.

$(s1) :> \; pleasant(x, y) \supset polite(y, x)$

$(s2) :> \; powerful(x) \supset polite(y, x)$

$\bullet\!> \; +pleasant: \;\; \langle Larry, Jack\rangle, \langle Peg, Ed\rangle, \langle Ramon, Killjoy\rangle,$
$\langle Srid, Alberto\rangle, \langle Alberto, Srid\rangle,$
$\langle Srid, Wlod\rangle, \langle Wlod, Srid\rangle$

$\bullet\!> \; +powerful: \;\; Bush, Gore, Cuomo, Gotti, Bonnie, Clyde$

•> $\neg polite$: $\langle Jack, Larry\rangle, \langle Ed, Peg\rangle, \langle Killjoy, Ramon\rangle, \langle Livia, Claudius\rangle$
•> $+polite$: $\langle Peg, Maggie\rangle, \langle Maggie, Peg\rangle, \langle Ollie, Kevin\rangle$
Here, as a shorthand for conjunctions of for-sure assertions of positive or negative literals, we listed the satisfying tuples. In addition, $\mathcal{A}_1$ includes the Uniqueness of Names as a for-sure axiom.

Then $\mathcal{A}_1$ is sympathetically solitary.

Theorem 12 (Explicit Solution, Sympathetic-Solitary)

Suppose that a prioritized default circumscription is sympathetically solitary cf. Definition 10. 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: Non-trivial. The argument involves substitution (of G for σQ) in the circumscription formula, properties of positivity / negativity, and lemmas about prioritization. See [Grosof, 1992] for the full details of the proof. □

Intuition: Predicates Q Pull Together Towards Bounds:
Theorem 12 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 12). 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.

Example 13 (Socializing, Explicit Solution)

Consider the Socializing axiom set $\mathcal{A}_1$ that we gave earlier in Example 11. Because the Socializing axiom set is sympathetically solitary, by Theorem 12, there is no conflict at all between the defaults, and there is an explicit solution to the Socializing circumscription. Corresponding to the pleasant-ness default ($s1$) is the **generating axiom** (recall section 3):

$$\forall x, y.\ \neg\{[(x{=}Larry) \wedge (y{=}Jack)] \vee [(x{=}Peg) \wedge (y{=}Ed)]\}$$
$$\wedge\ pleasant(x, y)$$

$$\supset \ polite(y,x)$$

I.e., the default "goes through" for every instance except for the two for-sure violations.

Corresponding to the powerful-ness default ($s2$) is the **generating axiom** (recall section 3):

$$\forall x, y.\ powerful(x) \supset polite(y,x)$$

For this default rule, there are no for-sure exceptions/violations.

Together with the base axioms, the above two generating axioms monotonic-logically entail (i.e., generate) *all* of the conclusions in the entire, *non*-monotonic Socializing PDC theory.

Observation 14 (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. 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. Example 15, below, illustrates.

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], 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 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.

Compared to **stratified logic programs with negation as failure**: sympathetic-solitary is plain different. One way to understand this is that: in any abnormality-style

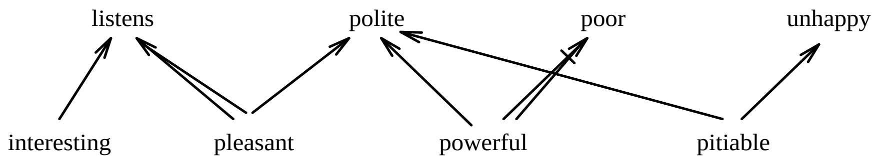

Figure 1: Shallow Topology of Sympathetically Solitary Class.

PPC that corresponds equivalently[6] to a sympathetic-solitary PDC, **only some, not all, predicates are being minimized**.

In our view, 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 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 11. Illustrated, in conventional inheritance net / semantic net style, is a simplified version of the base and default rules from the sympathetically solitary Example 11. 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.

Example 15 (Socializing, Limits of Sympathetically Solitary)

Each of the following axioms has the property that, considered as a single-axiom update to the axiom set in Example 11, it results in a new axiom set that is *not* sympathetically solitary.

$(s6) :> \ polite(x,y) \supset good\_tempered(x))$
$(s7) :> \ listens(x,y) \supset understands(x,y)$
$\bullet\!> \ \forall x.\, homeless(x) \supset pitiable(x)$
$\bullet\!> \ pleasant(Karen, Srid) \vee pleasant(Gus, Srid)$

Theorem 12 can also be applied to abnormality-style prioritized *predicate* circumscriptions.

Corollary 16 (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

$$B[ab, Q, Y, W] \ \equiv \ (\neg ab \leq D[Q, W]) \ \wedge \ (\sigma Q \leq G[W]) \ \wedge \ S[Q, Y, W]$$

or the form

$$B[ab, Q, Y, W] \ \equiv \ (\neg ab = D[Q, W]) \ \wedge \ (\sigma Q \leq G[W]) \ \wedge \ S[Q, Y, W]$$

[6]Recall Theorem 2 in section 2.

where D, σ, Q, G, and S are as in Theorem 12. Then

$$PPC(B; ab; R; fix\,W; ab, Q, Y, W) \equiv$$
$$B[ab, Q, Y, W] \wedge (\neg ab = D[Q, W]) \wedge (D[Q, W] = E[W])$$

where E is as in Theorem 12.

Proof : Immediate from Theorem 12 (default case) plus Theorem 2. □

Defaults Versus Minimized Predicates:
Note that solitary (and variable-solitary) is *not* a special case of this abnormality version of sympathetically solitary. Thus Corollary 16 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.

Observation 17 (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. 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:

2. **Rule Form and Explicit Solution:** In Theorem 12, 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 $GG[W] \stackrel{\text{def}}{=} \neg G[W]$.) 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).

3. **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 12 can be computed in time that is, worst-case, **polynomial** in the size of the CLD axiom set. In section 7, we discuss algorithms and computational complexity.

7 Algorithms and Tractability

We observe [7] next that the 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).

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

[7] Lack of space prevents us from going into much fuller details about algorithms and computational complexity analysis.

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, where also all functions are fixed, is as follows. (Recall from section 2 that domain closure and uniqueness of names, for example, imply that all functions are effectively fixed).

1. The first step is to scan the CLD axiom set for the fixtures. These are tested for: they should fix only predicates and (all the) functions. 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 a list σ of each predicate's sign of appearance, 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 10.

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.

Procedurally **performing** the explicit-solution 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 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.

New Special-Case Forward Inference Procedures:
The explicit solution yields generating axioms. (Recall the explanation in section 3 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 section 3 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 difficulty 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].

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, 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. This selectiveness can be at a variety of grain sizes: by individual default instances, by individual open defaults, or by sets of defaults. This can save computational effort.

8 Discussion

Additional Results: In [Grosof, 1992], we discuss how the explicit solution can be viewed at a finer grain-size: in terms of individual default instances. By a “default instance”, we mean a ground instantiation of a default formula. We also show there that belief revision is, in several respects, relatively easy for the sympathetically solitary case. By “belief revision”, we mean the problem of figuring out which previous conclusions need to be retracted after the CLD axiom set is updated with new premises/axioms.

Applicability to Other Non-Monotonic Formalisms: Using Observation 3 and the other mapping relationships discussed in section 4, it is straightforward to apply our results to several other non-monotonic formalisms, including the skeptical versions of Default Logic, Poole’s formalism, and Autoepistemic Logic.

9 Future Work

Next, we list several directions for future work.

In this paper, we did not (partly, due to constraints of space) go into full details about algorithms for sympathetically-solitary and their computational complexity analysis. We have not yet implemented this class, nor applied it to a serious problem.

Before embarking on such experimentation, it appears worthwhile to try to further expressively generalize the sympathetically solitary class, including to cases where there is some conflict but it is resolved completely by prioritization, say prioritization based on specificity dominance cf. default inheritance.

Another interesting direction to explore is the relationship of our explicit solution result (Theorem 12) to the recent results of Gabbay and Ohlbach [1992] about methods for eliminating second-order quantifiers.

References

[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.

[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.

[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.

[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.

[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.

[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*.

[Etherington, 1988] D. Etherington. *Reasoning with Incomplete Information*. Pitman, London, 1988.

[Gabbay and Ohlbach, 1992] Dov Gabbay and Hans Jurgen Ohlbach Ohlbach. Quantifier elimination in second-order predicate logic. *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR92)*, pages 425–435, San Mateo, CA, 1992. Morgan Kaufmann. Held Cambridge, MA, October 25–29, 1992.

[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.

[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, 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, 1992] Benjamin N. Grosof. *Updating and Structure in Non-Monotonic Theories*. PhD thesis, Computer Science Dept., Stanford University, Stanford, California 94305, October 1992. Published by University Microfilms, Inc.. Also available as Research Report from Stanford University Computer Science Dept. and/or IBM T.J. Watson Research Center.

[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.

[Konolige, 1988] K. Konolige. On the relation between default logic and autoepistemic logic. *Artificial Intelligence*, 35:343–382, 1988.

[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, 1985a] V. Lifschitz. Computing circumscription. *Proceedings IJCAI-85*, pages 121–127, Los Angeles, CA, 1985.

[Lifschitz, 1985b] Vladimir Lifschitz. Closed-world databases and circumscription. *Artificial Intelligence*, 27:229–235, 1985.

[Lifschitz, 1987] 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.

[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.

[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.

[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.

[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.

[Shoham, 1988] Y. Shoham. *Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence.* MIT Press, Cambridge, Mass., 1988.

[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.

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

[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.