A SURVEY OF COMPLEXITY RESULTS FOR NON-MONOTONIC LOGICS*

MARCO CADOLI AND MARCO SCHAERF[†]

▷ This paper surveys the main results appearing in the literature on the computational complexity of non-monotonic inference tasks. We not only give results about the tractability/intractability of the individual problems but we also analyze sources of complexity and explain intuitively the nature of easy/hard cases. We focus mainly on non-monotonic formalisms, like default logic, autoepistemic logic, circumscription, closed-world reasoning, and abduction, whose relations with logic programming are clear and well studied. Complexity as well as recursion-theoretic results are surveyed. ◁

1. INTRODUCTION

Non-monotonic logics and negation as failure in logic programming have been defined with the goal of providing formal tools for the representation of *default* information. One of the ideas underlying both areas is that the use of default assumptions should lead to a more compact representation of knowledge. As a consequence, non-monotonic knowledge bases should be more space-effective than ordinary ones and this may hopefully have an impact on the performances of theorem provers.

This expectation generated an interesting activity in the development of algorithms for reasoning under non-monotonicity, as well as many studies about the inherent complexity of the inference tasks.

* Work partially supported by the ESPRIT Basic Research Action N. 6810 COMPULOG II and the Progetto Finalizzato Informatica of the CNR (Italian Research Council). The first author is supported by a CNR scholarship.

[†]Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza." Email: schaerf@assi.ing.uniroma1.it.

Address correspondence to Marco Cadoli, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza," via Salaria 113, 00198 Roma, Italia. Email: cadoli@assi.ing.uniroma1.it.

Received June 1992; accepted May 1993.

THE JOURNAL OF LOGIC PROGRAMMING

655 Avenue of the Americas, New York, NY 10010 0743-1066/93/$6.00

The goal of this paper is to survey the main results appearing in the literature on the computational complexity of non-monotonic inference tasks. We not only give results about the tractability/intractability of the individual problems, but we also analyze sources of complexity and explain intuitively the nature of easy/hard cases. Furthermore, we create an extensive list of references to the literature that, in our opinion, will provide a useful tool for researches interested in designing algorithms for non-monotonic reasoning (NMR in the sequel).

We focus mainly on non-monotonic formalisms, like default logic, autoepistemic logic, circumscription, closed-world reasoning, and abduction, whose relations with logic programming are clear and well studied. Other reasoning problems, like reasoning on inheritance networks or belief revision, are only briefly mentioned.

We refer the reader to works like [39] and [89] for extensive surveys on NMR and to [123] for a survey on the relations between NMR and logic programming. Works on NMR that survey complexity results have been done by Minker [103] and Schlipf [139].

Many results that have appeared in the literature are concerned with decidable fragments of non-monotonic logics. In the spirit of a well-established trend in knowledge representation [20], they aim at the characterization of the expressive power of languages having polynomial time reasoning procedures. Other works deal with fully expressive languages and try to characterize the precise complexity of the inference task. In this paper we are interested in both kinds of results, privileging the analysis of propositional, decidable languages.

We use the jargon of computational complexity and recursion theory, as found in [54] and in [131], respectively. In particular, we make use of the notions of polynomial, arithmetical, and analytical hierarchies.

Using notions of higher-order complexity is necessary, since a general property of non-monotonic inference is that its computational complexity is higher than the complexity of the underlying monotonic logic. As an example, restricting the expressiveness of the language to Horn clauses allows for polynomial inference as far as classical propositional logic is concerned [40], but the inference task becomes NP-hard when propositional default logic [150] or circumscription [26] are considered.

The fact that non-monotonicity adds complexity to reasoning was clear from the first studies: In his seminal paper, Reiter [129] showed that inference in default logic is not recursively enumerable (r.e. in the sequel). The issue of determining precise lower and upper bounds was addressed later, and some formalisms have been proved to be complete for precise levels of the arithmetical or analytical hierarchy [5, 25, 34, 135]. Analogous completeness results have been found for the propositional versions of the same formalisms wrt some levels of the polynomial hierarchy [41, 64].

Part of the increase in the complexity of inference can be explained by noticing that the semantic definitions of most non-monotonic formalisms are either based on fixpoint constructions or on conditions requiring some form of *minimality*. This apparently gives a completely orthogonal source of complexity.

Examples of formalisms based on fixpoint semantics are default [129] and autoepistemic logic [104] where deduction is performed wrt *extensions* or *expansions*, which are solutions of fixpoint equations. As we see in the following sections, this fixpoint construction requires an additional nondeterministic choice that cannot be polynomially reduced to a deterministic one unless the polynomial hierarchy collapses.

An example of non-monotonic formalism based on a form of minimality is circumscription [85, 98, 99], in which inference is performed wrt the models of a first-order formula in which the extension of some selected predicates is minimized. As it turns out, the minimality requirement gives rise to a computational overhead that is analogous to that given by the fixpoint construction.

The paper is organized as follows: After an introductory section on complexity concepts (Section 2), we first survey the formalisms based on fixpoint constructions, starting with default logic, which is presented in Section 3. The structure of reasoning in default logic makes it easy to point out precisely the various sources of complexity present in this kind of non-monotonic formalisms. We then discuss in Section 4 modal non-monotonic logics, with particular attention to autoepistemic logic. The introduction of negation in logic programming has produced a proliferation of semantics, some based on fixpoint constructions (e.g., stable model semantics) and other on minimal models (e.g., well-founded semantics). Complexity results for the various semantics are surveyed in Section 5. In Section 6 we discuss non-monotonic formalisms whose semantics is based on some form of minimality, which are the different forms of closed-world reasoning and circumscription. In Section 7 we carry on an analysis of the complexity of the so-called logic-based abduction and, very briefly, mention results for other forms of abduction. While abduction is not strictly a formalism for NMR, nevertheless it is tightly related both to it and to logic programming. In Section 8 we discuss in some detail the various reductions between NMR problems pointing out their importance from the point of view of computational complexity analysis. Finally, in Section 9 we draw some conclusions.

2. COMPLEXITY CLASSES

In this section we give a brief overview of complexity concepts that will be used throughout the paper. We refer the reader to [54] and [74] for a thorough introduction to the field of complexity.

In this paper we deal most of the time with *decision problems,* that is, problems that admit a boolean answer. For decision problems the class P is the set of problems that can be answered by a Turing machine in polynomial time. Often we refer to computations done by nondeterministic Turing machines. The class of decision problems that can be solved by a nondeterministic Turing machine in polynomial time—where it is understood that the answer is **yes** provided *at least one* of the computations done in parallel by the machine ends in an accepting state —is denoted by NP. The class of problems whose answer is always the complement of those in NP is denoted by co-NP. Also problems in co-NP can be solved by a nondeterministic Turing Machine in polynomial time, but it is understood that the answer is **yes** provided *all* the computations done in parallel by the machine end in an accepting state. The class P is obviously contained both in NP and in co-NP.

An example of a problem in NP is testing satisfiability of a propositional formula: A formula T is satisfiable iff *at least one* truth assignment M such that $M \models T$ exists. An example of a problem in co-NP is testing if a propositional formula T entails a propositional formula γ: $T \models \gamma$ iff *for all* truth assignments M it holds that $(M \models T) \Rightarrow (M \models \gamma)$. In fact propositional satisfiability (entailment) is an NP-*complete* (co-NP-*complete*) problem, that is the "toughest"—wrt many-one polynomial reducibility—problem in the class NP (co-NP). We recall that the best algorithms known for solving NP-complete or co-NP-complete problems require exponential time in the worst case, and that the following relations are conjectured: $P \subset NP$, $P \subset$ co-NP, $NP \neq$ co-NP.

In the following we refer to a particular type of computation called computation with *oracles*. Oracles are intuitively subroutines without cost. Given a class of decision problems C, the class P^C (NP^C) is the class of decision problems that can be solved in polynomial time by a deterministic (nondeterministic) machine that uses an oracle for the problems in C, that is a subroutine for any problem in C that can be called several times, spending just one time-unit for each call.

The definition of *polynomial hierarchy* is based on oracle computations. The classes Σ_k^p, Π_k^p, and Δ_k^p of the polynomial hierarchy are defined by

$$\Sigma_0^p = \Pi_0^p = \Delta_0^p = P$$

and, for all $k \geq 0$,

$$\Sigma_{k+1}^p = NP^{\Sigma_k^p}, \qquad \Pi_{k+1}^p = \text{co-}\Sigma_{k+1}^p, \qquad \Delta_{k+1}^p = P^{\Sigma_k^p}.$$

Notice that $\Sigma_1^p = NP$, $\Pi_1^p =$ co-NP, $\Delta_1^p = P$. The following relations have been conjectured:

$$\Delta_k^p \subset \Sigma_k^p, \qquad \Delta_k^p \subset \Pi_k^p \qquad \Sigma_k^p \subset \Delta_{k+1}^p, \qquad \Pi_k^p \subset \Delta_{k+1}^p, \qquad \Sigma_k^p \neq \Pi_k^p,$$

for all $k \geq 1$.

We say that a problem is at the kth level of the polynomial hierarchy if it is Δ_{k+1}^p-complete under polynomial Turing reductions, that is, it is in Δ_{k+1}^p and it is either Σ_k^p-hard or Π_k^p-hard. Propositional satisfiability and entailment are both at the first level.

In the following we focus on problems in Σ_2^p ($= NP^{NP}$) and Π_2^p ($=$ co-NP^{NP}), as they are important in this paper. Σ_2^p contains all problems solvable in nondeterministic polynomial time provided it is possible to use for free a subroutine for a problem in NP—for example, propositional satisfiability. The prototypical Σ_2^p-complete problem is testing validity of a quantified boolean formula Q (called $QBF_{2,\exists}$ formula) of this kind:

$$\exists x_1 \cdots \exists x_n \forall y_1 \cdots \forall y_m E$$

where $x_1, \ldots, x_n, y_1, \ldots, y_m$ are distinct propositional letters and E is a purely propositional formula built on such letters. The formula Q is valid iff there exists a truth value assignment to the propositional letters $x_1, \ldots, x_n$ such that for each possible extension of such an assignment to the letters $y_1, \ldots, y_m$, Q is true. The prototypical Π_2^p-complete problem is testing validity of a quantified boolean formula of this kind (called a $QBF_{2,\forall}$ formula):

$$\forall x_1 \cdots \forall x_n \exists y_1 \cdots \exists y_m E$$

We refer the reader to [111] for examples of Δ_2^p-complete problems.

In virtue of the conjectures $\Sigma_k^p \subset \Sigma_{k+1}^p$ and $\Pi_k^p \subset \Pi_{k+1}^p$, Σ_2^p-complete and Π_2^p-complete problems are considered more difficult to solve than both NP-complete and co-NP-complete problems. A practical difference exists between NP-complete and Σ_2^p complete problems. Suppose we have a good heuristic for solving an NP-complete problem (as an example, propositional satisfiability testing) that normally solves the problem in an acceptable amount of time—even if exponential time is needed in the worst case. It is still not immediate how to use such a good heuristic for solving efficiently a Σ_2^p-complete problem. As a matter of fact, if the conjecture $NP \subset \Sigma_2^p$ is true, then in the worst case an exponential number of calls to an algorithm for satisfiability testing would be necessary in order to solve any Σ_2^p-complete problem. The same holds for Π_2^p-complete problems. Methods such as GSAT [145] for efficient handling of NP-complete problems are therefore most likely not applicable to problems at the second level of the polynomial hierarchy. On the other hand Δ_2^p-complete ($= P^{NP}$-complete) problems are "mildly" harder than NP-complete ones, since they can be solved by means of a polynomial number of calls to an algorithm for satisfiability testing.

As we shall see in the sequel, many problems in propositional non-monotonic reasoning are Σ_2^p-complete or Π_2^p-complete, and many algorithms designed for solving such problems use satisfiability testers as subroutines.

The last class of decision problems we introduce are those polynomially solvable by a deterministic Turing machine with no more than $f(n)$ calls to a Σ_k^p oracle. Such a class is denoted by $P^{\Sigma_k^p[f(n)]}$, where $f(n)$ is a polynomial function of the size n of the problem instance. In particular we will mention in the paper the class $P^{NP[\log(n)]}$. We notice that $NP \subseteq P^{NP[\log(n)]} \subseteq \Delta_2^p$ and that the containments are conjectured to be strict.

Throughout this paper we assume that all of the above mentioned conjectures are true.

If two decision problems A and B are complete for the same class, then there is always a way to solve any instance of A by solving a single instance of B and vice versa. In Section 8 we will discuss in detail applications of this property to NMR problems.

Sometimes we refer to *search* problems, that is, problems whose answer is more complex than just a boolean value. As an example, *finding* a satisfying truth assignment for a propositional formula is a search problem. An interesting class of search problems is $F\Delta_2^p$, which is the set of problems solvable in polynomial time by a machine with access to an oracle for an NP problem. If the oracle can be accessed only a logarithmic number of times, then we have the class $FP^{NP[\log(n)]}$.

While the polynomial hierarchy is an attempt at characterizing "how polynomially uncomputable" is a decidable function, the arithmetical and analytical hierarchies—which are precursors of the polynomial hierarchy—are characterizations of "how undecidable" is a function. For the sake of brevity we do not give precise definitions but refer the reader to [131], and [70]. We just recall that the elements $\Delta_k^0, \Sigma_k^0, \Pi_k^0$ of the arithmetical hierarchy and the classes $\Delta_k^1, \Sigma_k^1, \Pi_k^1$ of the analytical hierarchy are defined similarly to the elements of the polynomial hierarchy. While separation among the levels of the polynomial hierarchy is only conjectured, separation in both the arithmetical and the analytical hierarchies has been formally proved. Moreover, any element of the analytical hierarchy is harder than any element of the arithmetical hierarchy.

3. DEFAULT LOGIC

Default logic has been defined by Reiter in [129] and it is one of the more extensively studied non-monotonic formalisms. Interesting relations between default logic and logic programming have been shown by Bidoit and Froidevaux in

[13]. They used default logic for defining a semantics for negation in logic programming.

In default logic the knowledge about the world is divided into two parts, representing certain knowledge and defeasible rules, respectively. The first part (denoted by W) is a set of closed first-order formulae, while the second one (denoted by D) is a collection of special inference rules called *defaults*. A default is a rule of the form

$$\frac{\alpha(\mathbf{x}) : \beta_1(\mathbf{x}), \ldots, \beta_n(\mathbf{x})}{\gamma(\mathbf{x})},$$

where $\alpha(\mathbf{x}), \beta_1(\mathbf{x}), \ldots, \beta_n(\mathbf{x}), \gamma(\mathbf{x})$ are wffs whose free variables are among those of $\mathbf{x} = x_1, \ldots, x_m$. $\alpha(\mathbf{x})$ is called the *prerequisite* of the default, $\beta_1(\mathbf{x}), \ldots, \beta_n(\mathbf{x})$ $(n \geq 0)$ are called *justifications*, and $\gamma(\mathbf{x})$ is the *consequence*. When $n = 0$, then the propositional constant **true** is implicitly assumed as the justification of the default. A default is *closed* if none of α $\beta_1, \ldots, \beta_n$, γ contains free variables. A default theory $\langle D, W \rangle$ is closed iff all the defaults in D are closed.

There are two special cases of default rules which are frequently used. Default rules of the form $\alpha(\mathbf{x})$: $\beta(\mathbf{x})/\beta(\mathbf{x})$ and $\alpha(\mathbf{x})$: $\beta(\mathbf{x}) \wedge \gamma(\mathbf{x})/\beta(\mathbf{x})$ are called *normal* and *seminormal*, respectively.

The semantics of a closed default theory $\langle D, W \rangle$ is based on the notion of *extension*, which is a possible state of the world according to the knowledge base. Formally, an extension is a fixed point of the operator Γ defined as follows. $\Gamma(A)$ is the smallest set such that the following hold:

1. $W \subseteq \Gamma(A)$;
2. $\Gamma(A) = \{\alpha \mid \Gamma(A) \models \alpha\}$;
3. if $(\alpha : \beta_1, \ldots, \beta_n/\gamma) \in D$, $\alpha \in \Gamma(A)$, and $\forall_{j=1}^{n}$ $\neg \beta_j \notin A$, then $\gamma \in \Gamma(A)$.

A set of formulae E is an extension of $\langle D, W \rangle$ iff $E = \Gamma(E)$. This definition is extended to *open* default theories by assuming that the defaults with free variables implicitly stand for the infinite set of closed defaults obtained by replacing the free variables with terms of the Herbrand universe of the default theory.

The three computational problems that are most relevant in default logic, and that have been extensively studied in the literature, are deciding whether a default theory $\langle D, W \rangle$ has an extension, deciding whether a formula α belongs to at least one extension of $\langle D, W \rangle$ (also known as *credulous* default reasoning), and deciding whether α belongs to all the extensions (*skeptical* default reasoning). Notice that these are decision problems.

Each extension E of $\langle D, W \rangle$ is identified by a subset D' of D, called the set of *generating defaults* of E, having the property that E is the deductive closure of

$$W \cup \{\gamma \mid (\alpha : \beta_1, \ldots, \beta_n/\gamma) \in D'\}.$$

The set of generating defaults gives a compact representation of an extension of a closed default theory, which by definition is a deductively closed set of formulae, hence infinite. The *search* problem of finding a set of generating defaults has also been studied from the computational point of view.

We would like to explain precisely the sources of complexity of default logic, focusing on the propositional case. A simple algorithm for deciding whether a formula δ follows by credulous default reasoning from $\langle D, W\rangle$ is as follows:

```
for each subset D' of D:
do begin
    (* D' is potentially a set of generating defaults *)
    if D' corresponds to an extension of ⟨D,W⟩ (* 1st test *)
    then begin
        W' := W ∪ {γ|(α: β_1,...,β_n/γ) ∈ D'}.
        if W' ⊨ δ (* 2nd test *)
        then return true
      end;
  end;
return false.
```

Since there are $(2^{|D|})$ subsets of D, the body of the **for each** loop could be executed an exponential number of times. As far as the computational cost of the body of the loop is concerned, the first test—according to the definition of extension given previously—requires us to perform several satisfiability and entailment checks for propositional formulae. Analogously, the second test requires an entailment check. As a consequence, each execution of the body of the loop requires most likely an exponential time.

If nondeterministic computations are considered, then the situation is different. As an example, it is possible to guess in parallel all the subsets D' of D, and this can be done in polynomial time by a nondeterministic machine. Clearly we want such a machine to return **true** if *at least one* subset D' satisfies the property in the loop. As far as the body of the loop is concerned, we saw in Section 2 that both satisfiability and entailment tests can be answered by NP machines. In other words, credulous default reasoning can be done by a nondeterministic polynomial Turing machine which guesses in parallel all the possible subsets of D' of D and, for each subset, uses the answers given by an *oracle*. The oracle is in charge of performing the satisfiability and entailment checks required by the body of the loop, which are computations doable in nondeterministic polynomial time as well. It is clear from the definition of extension that for each subset D' only a polynomial number of calls to the oracle are needed.

The above argument can be made more formal and it shows that the problem of credulous default reasoning is in the class $\mathrm{NP}^{\mathrm{NP}} = \Sigma_2^p$ of the polynomial hierarchy. This establishes an upper bound, but it is of obvious practical interest to know whether (at least one of) the nondeterministic polynomial computations that we were referring to previously can be turned into deterministic polynomial ones, by means of smart search techniques.

Gottlob in [64] and, independently, Stillman in [151] give a negative answer to this problem, by showing that credulous reasoning is Σ_2^p-complete, that is, the hardest problem among those of the class Σ_2^p.

This result can be interpreted by saying that the source of complexity of consistency checking and that of the choice of the generating defaults are independent, and their interaction gives a problem which is complete for the second level of the polynomial hierarchy. This fact formalizes, for the propositional case, the intuition that, if NP $\neq \Sigma_2^p$, default reasoning is computationally harder than monotonic reasoning, which is complete for the first level of the polynomial hierarchy.

It is interesting to see explicitly Gottlob's proof of Σ_2^p-hardness, in order to understand better the sources of complexity. The proof consists in a polynomial reduction from the problem of deciding the validity of a $QBF_{2,\exists}$ formula.

Let $Q = \exists p_1 \cdots \exists p_n \forall q_1 \cdots \forall q_m G$ be an arbitrary $QBF_{2,\exists}$ formula. Let $\langle D, \emptyset \rangle$ be a default theory, where

$$D = \{(: p_1/p_1), (: \neg p_1/\neg p_1), \ldots, (: p_n/p_n), (: \neg p_n/\neg p_n)\}.$$

All the possible truth assignments to $p_1, \ldots, p_n$ are in one-to-one correspondence with the extensions of $\langle D, \emptyset \rangle$. Hence Q is valid iff there exists at least one extension of $\langle D, \emptyset \rangle$ containing G, that is, if G follows from $\langle D, \emptyset \rangle$ in a credulous way. This shows that validity of a $QBF_{2,\exists}$ formula can be reduced to credulous default reasoning by means of a polynomial many-one reduction.

We want to spend few words commenting on the above reduction. Extensions can be chosen in nondeterministic polynomial time (symbolized by the sequence of the existential quantifiers), but apparently not in deterministic polynomial time. An intuitive explanation of this fact is that defaults may interact in a combinatorial fashion, thus generating an exponential number of extensions. Moreover, even if a deterministic polynomial choice were possible, the complexity of the inference wrt the chosen extension should still be faced: it can be done in nondeterministic polynomial time (sequence of universal quantifiers), but apparently not in deterministic polynomial time.

The above proof uses normal defaults without prerequisites. As a consequence, restricting to normal default theories does not help reduce the complexity of credulous reasoning. From the algorithmic point of view, normal default theories have been considered more promising, since they support goal-directed algorithms for credulous reasoning (see [129]), while the algorithms proposed for non-normal theories by Etherington in [50] and Zhang and Marek in [158] use an exponential amount of space to avoid nontermination.

Skeptical default reasoning in the prerequisite-free normal case has been shown to be Π_2^p-complete with a similar proof in [64] and [151]. In the same works, the problem of showing the existence of an extension has been proven to be Σ_2^p-complete for seminormal default theories (it becomes trivial for normal default theories which are guaranteed to have extensions). Papadimitriou and Sideri prove in [113] similar results for the last problem.

Several researchers studied restrictions of the expressiveness of default theories so that inference could be done in polynomial time. In particular restricted forms of the purely propositional part (the W), like Horn clausal form, 2CNF or 1CNF, have been considered (2CNF formulae are conjunction of clauses of the form $x \vee y$, where x, y are literals; 1CNF formulae are just conjunction of literals). Generally speaking, restricting the expressiveness of the W is not useful from the computational point of view, since the complexity of default reasoning can be completely "hidden" in the default part: the default theories $\langle D, W \rangle$ and $\langle D \cup \{(: /W)\}, \emptyset \rangle$ are equivalent. Therefore, the expressiveness of the default part must be limited as well, so that the nondeterministic choice of the subsets of D that are to form a set of generating defaults can be turned into a deterministic one. This can be done by imposing conditions on the syntactic form of the defaults in order to control their interaction.

Let us analyze one such restriction that has been proposed, the so-called *Horn* default. In Horn defaults the prerequisite is a conjunction of positive atoms, the justification and the consequence are the same literal, that is the default has the form

$$\alpha : y/y$$

Kautz and Selman [76] prove that, as long as W is 1CNF and credulous reasoning is concerned, a set of Horn defaults D can be mapped in linear time into a set of Horn clauses H so that the inference of a single literal x from $\langle D, W \rangle$ is equivalent to the inference of x from H, hence doable in linear time in $|\langle D, W \rangle|$. In this case clearly the interaction among the defaults is not arbitrarily complex and can be easily controlled. As proven by Stillman [150], the hypothesis that W is 1CNF cannot be fully relaxed, since if W is in Horn clausal form, then credulous reasoning is NP-complete.

In [76] and [150] other examples of restrictions that lead to polynomially tractable cases are shown. Kautz and Selman [76] focus on default theories in which W is 1CNF, finding results for all three decision problems and the search problem of default reasoning. It is interesting to notice that credulous and skeptical reasoning have different complexity, for example, skeptical reasoning of a single literal is co-NP-complete when W is 1CNF and D is Horn.

Finding a set of defaults generating an extension has still a different complexity, and it is polynomial for some class in which both skeptical and credulous reasoning are intractable. Kautz and Selman, for example, analyze the case in which W is 1CNF and D has only *disjunction-free ordered* defaults, which are strictly more expressive than Horn defaults. Orderedness is a property of defaults defined by Etherington [51], analogous to the idea of stratification in logic programming. A disjunction-free default has the form $a_1 \wedge \cdots \wedge a_l : b_1 \wedge \cdots \wedge b_m \wedge c_1 \wedge \cdots \wedge c_n / b_1 \wedge \cdots \wedge b_m$, where all a_i, b_j, and c_k are literals. Finding an extension in this case has been proven [76] to be polynomial. This result has been improved by Papadimitriou and Sideri [113], who show that some form of cycles in the defaults can be allowed. In particular, orderedness can be relaxed to a property that they call *evenness*, still having polynomiality.

It is clear that finding a set of defaults generating an extension is at least as hard as proving the existence of an extension. The last problem is shown [76] to be NP-complete when W is 1CNF and D disjunction-free. This result is improved by Dimopoulos and Magirou [38], in which NP-completeness is shown for disjunction-free and prerequisite-free defaults.

Kautz and Selman give also upper bounds for the complexity of default reasoning, showing that as long as W is 1CNF and D is disjunction-free (the most general class they take into account) credulous reasoning and proving the existence of an extension are in NP, skeptical reasoning is in co-NP and finding an extension is in $F\Delta_2^p$, that is, it can be done by a polynomial machine which uses an NP oracle. These results confirm the intuitive complexity analysis that we gave before: The complexity of the consistency/entailment checking is always linear in 1CNF-disjunction-free default theories, hence the only source of complexity is the interaction of defaults. As a consequence, the decision problems are at the first level of the polynomial hierarchy.

Stillman [150] tries to enhance the expressiveness of W, by allowing either 2CNF or Horn clauses, with disjunction-free defaults. For the reasons given before, the upper bound of the four problems is not affected by this enhancement. He focuses on prerequisite-free defaults, showing for example that if the defaults are also normal and the W is 2CNF, then credulous reasoning is polynomial, while it is NP-complete if W is in Horn clausal form.

Bidoit and Froidevaux [13] give results that complement Stillman's, by showing that propositional logic programs can be translated into default theories whose W is Horn. If the program is stratified, then the resulting default theory has a unique extension which can be computed in polynomial time. Further polynomial cases can be obtained using the similar translations by Marek and Truszczyński [94].

Dimopoulos and Magirou [38] obtain further results, by using a network representation of seminormal disjunction-free default theories. Using a graph-theoretic analysis they show that for theories without odd cycles in the associated graph, finding an extension can be done in polynomial time. This class generalizes the class of disjunction-free ordered theories, thus improving the previous results in [76].

Ben-Eliyahu and Dechter [9, 10] propose a different technique for performing default reasoning: translating a default theory $\langle D, W \rangle$ into a propositional theory $\mathscr{P}$ so that there is a one-to-one correspondence between the models of $\mathscr{P}$ and the extensions of $\langle D, W \rangle$. As proven by the results of [64] and [151], a transformation of this kind cannot in general be done in polynomial time unless $\Sigma_2^p = \text{NP}$. Actually the size of $\mathscr{P}$, according to the transformation of [10], is exponential in the size of $\langle D, W \rangle$. Nevertheless, there are subcases in which the size of $\mathscr{P}$ is polynomial wrt the size of the default theory. The most important such case shown in [10] concerns the class of *2-default theories*, in which W is 2CNF, the prerequisite of each default is in 2CNF, each justification is in 2DNF, and the conclusion is a clause of size 2. Using this reduction, Ben-Eliyahu and Dechter prove that the three decision problems are either NP-complete or co-NP-complete for this class of default theories. Furthermore, each subclass of 2-default theories that translates into a tractable subclass of propositional satisfiability is a tractable subset of default logic. Two such subclasses are shown in [10].

With respect to default theories in which first-order formulae occur, it is worth remembering that a default of the form $(\,:\, p(x)/p(x))$ where x is a variable, is actually an abbreviation for the infinite set of defaults $(\,:\, p(t)/p(t))$, where t is a term of the Herbrand universe of the default theory.

Apart from the results by Reiter [129], who showed that credulous default reasoning is not r.e., few results on the complexity of fully first-order default reasoning have appeared. Reiter proved also that, as long as closed normal default theories built on decidable fragments of first-order logic are concerned, decidability of credulous reasoning is guaranteed. When open defaults are considered, this is no longer true: Baader and Hollunder [7] showed that credulous default reasoning is undecidable for default theories built on a decidable fragment of first-order logic and containing a finite number of open defaults. Apt and Blair [5] performed an analysis on complexity of negation in logic programs, giving some results on default logic by using the existing correspondences between the two formalisms. In particular, they show that skeptical and creduluous inference of ground atoms is, for all finite n, hard wrt the class Σ_n^0 of the arithmetic hierarchy, hence strictly harder than first-order inference. These results hold for finite first-order W and for infinite D, in the sense specified above. In Section 5 we give more details about the results of Apt and Blair.

Marek, Nerode, and Remmel [91] give recursion-theoretic results on a non-monotonic formalism that generalizes default logic.

We now briefly mention results on variants of default logic. The complexity of model-preference default logic, introduced by Kautz and Selman [142], has been analyzed by the same authors as well as Papadimitriou [112] and Cadoli [24]. The complexity of another variant, known as default logic with stationary extensions introduced by Przymusinksa and Przymusinsky [120], has been analyzed by Gottlob [63].

A computationally appealing alternative to default logic is based on the use of probabilistic semantics. For a general survey on the topic we refer the reader to Pearl's paper [116]. In this setting, default rules implicitly define a preference ordering (ranking) among sentences that is used to infer conclusions. This different reading has a stronger causal content than classical default logic. Computationally, this results in a more tractable system. In fact, the problem of deciding whether a formula follows from a conditional knowledge base is in Δ_2^p. This is due to the simpler interaction between defaults which can be accommodated using only a polynomial number of calls to a satisfiability tester. As a consequence, restricting the attention to base languages where satisfiability can be checked in polynomial time (e.g., Horn clauses) delivers polynomial deduction tasks (see the works of Goldszmidt and Pearl [62]).

We would like to conclude this section by summarizing the results previously presented. Propositional default reasoning is in some formal sense strictly harder than classical propositional reasoning. Apparently it has two independent sources of complexity (selection of an appropriate set of generating defaults, and consistency/entailment checks) whose interaction leads to problems complete for the second level of the polynomial hierarchy. Many successful attempts at finding polynomially tractable subcases have been proposed, and all of them attack both sources of complexity at the same time.

Algorithms for performing default reasoning have been presented by Reiter [129], Etherington [50], Zhang and Marek [158], Schwind [141], and Poole et al. [117]. Algorithms described in [129], [50], and [158] use satisfiability testers as subroutines.

4. MODAL NON-MONOTONIC LOGICS

In this section we review complexity results presented in the literature concerning non-monotonic versions of modal logics, with special attention to the best known of such logics, autoepistemic logic with all its variants. We consider a propositional language augmented by the modal operator L.

Historically, the first non-monotonic modal logic was the Non-monotonic Logic (NM1 in the sequel) introduced by McDermott and Doyle [101, 100]. This logic has been refined by Moore [104], who defined autoepistemic logic. In autoepistemic logic the semantics of a set of modal formulae A (called *premises*) is given by *stable expansions* which are fixpoints of the following equation:

$$\Delta = \{ \alpha \mid A \cup L\Delta \cup \neg L\overline{\Delta} \models \alpha \},$$

where $L\Delta = \{L\beta \mid \beta \in \Delta\}$ and $\neg L\overline{\Delta} = \{\neg L\beta \mid \beta \notin \Delta\}$.

Stable expansions in autoepistemic logic have the same role played by extensions in default logic. Analogously to default logic, the main reasoning tasks are deciding if a set of premises has a stable expansion, if a formula α belongs to at least one stable expansion of a set of premises (credulous autoepistemic reasoning), and if a formula belongs to all stable expansions of a set of premises (skeptical autoepistemic reasoning).

There is a close correlation between the interpretation of $\neg L\alpha$ in autoepistemic logic and the meaning of *not* α in logic programming, where *not* denotes negation as failure. This analogy has been stressed and used by Gelfond [56] to define the iterative fixed-point semantics and by Gelfond and Lifschitz [57] to define the stable model semantics for logic programs with negation.

An issue of obvious computational interest concerns the compact representation of stable expansions, which is analogous to the issue of a compact representation of an extension of a default theory. As shown by Konolige [79], any set of premises A can be transformed into an equivalent one A' which is composed of formulae f_i of the kind

$$L\alpha_i \wedge \neg L\beta_i^1 \wedge \cdots \wedge \neg L\beta_i^n \rightarrow \gamma_i,$$

where $\alpha_i, \beta_i^1, \ldots, \beta_i^n, \gamma_i$ are formulae not containing the modal operator L. When formulae are in this syntactic form, then subsets of A' can play the same role that generating defaults have in default logic. Unfortunately, the process of transforming arbitrary formulae into this form may exponentially increase their size. For this reason Niemelä [108] and Shvarts [147] have proposed methods which extract directly from A *kernels*, that is, sets of subformulae of A which are in one-to-one relation with stable expansions.

The complexity results for the general case are the following. The problem of deciding whether a set of premises A has a stable expansion is Σ_2^p-complete, credulous autoepistemic reasoning is also Σ_2^p-complete, while skeptical autoepistemic reasoning is Π_2^p-complete. The upper bounds have been proven by Niemelä [108] and the lower bounds by Gottlob [64].

Notice that these results are exactly the same as those already shown for default logic. Since all problems complete for the same class of the polynomial hierarchy can be polynomially reduced one to the other, it is clear that, for example, credulous default reasoning is polynomially reducible to credulous autoepistemic reasoning. More precisely, the existence of a polynomial mapping $\mathcal{M}$ from the problem of deciding credulous inference of a formula γ from a default theory $\langle D, W\rangle$ into the problem of credulous inference of another formula δ from a set of premises A is guaranteed. The existence of an inverse mapping $\mathcal{M}'$ is guaranteed as well.

This may seem in contradiction with the negative results obtained by Konolige [79], and by Marek and Truszczyński in [95], where they could not find an embedding of default logic in classical autoepistemic logic, but only in some of its variants. This contradiction is only apparent; in fact they were interested in translations satisfying some condition of locality, in which a single default rule is mapped into a modal formula. Transformation like $\mathcal{M}, \mathcal{M}'$ do not necessarily satisfy locality. A more detailed analysis of the various translations proposed for different NMR formalisms is presented in Section 8.

As in the case of default reasoning, restricting the expressiveness of the language may lead to a decrease of the complexity. For example, Marek and

Truszczyński have shown [97] that deciding whether an atom is member of one expansion of a set of premises of the form:

$$La_1 \wedge \cdots \wedge La_n \wedge \neg Lb_1 \wedge \cdots \wedge \neg Lb_m \rightarrow c,$$

where $a_1, \ldots, a_n, b_1, \ldots, b_m, c$ are literals, is NP-complete while membership to all expansions is co-NP-complete. In a recent paper Niemelä and Rintanen [109] show that restricting the attention to stratified autoepistemic theories, a class defined by Marek and Truszczyński [97] in the style of stratified logic programs, it is possible to greatly reduce the computational complexity of all the decision problems. In particular, they show that for stratified autoepistemic theories the only source of complexity is given by the satisfiability checking in the underlying propositional language; hence, when this check can be done in polynomial time (e.g., Horn clauses), then all the decision problems are polynomially tractable.

Although there are no other direct results on the complexity of autoepistemic inference, other complexity results can be obtained by means of the techniques described in Section 8, using the translations from default logic [65, 87] and logic programs [57, 93] into autoepistemic logic.

In the rest of this section we deal with several variants of autoepistemic logic that have been proposed in the literature.

A more complex variant is the *moderately grounded* autoepistemic logic introduced by Konolige [79]. A stable expansion T is moderately grounded if and only if there is no other stable expansion T' such that the part of T' with no occurrence of L is strictly contained in that of T. This minimality requirement for T can be checked by nondeterministically selecting a T' and checking whether the condition holds, but this obviously adds one further level of nondeterminism to the computation. Eiter and Gottlob [46] show that it cannot be eliminated by proving that credulous reasoning wrt moderately grounded expansions is Σ_3^p-complete and skeptical reasoning is Π_3^p-complete.

Autoepistemic logic is a refinement of the non-monotonic logic NM1 introduced by McDermott and Doyle [101, 100]. NM1 is based on a fixpoint construction slightly different from the one used in autoepistemic logic. Given the analogies of NM1 with autoepistemic logic, it is not surprising that the decision problems of NM1 have the same complexity of the corresponding problems of autoepistemic logic, see Niemelä [108] (upper bounds) and Gottlob [64] (lower bounds).

Using the fixpoint construction of McDermott and Doyle but changing the underlying modal system, Marek and Truszczyński [95] and Marek, Shvarts, and Truszczyński [92] present a whole new family of non-monotonic modal systems. The complexity analysis of these systems has shown that for most of them all the decision problems lie at the second level of the polynomial hierarchy (Σ_2^p- or Π_2^p-complete) even when underlying modal systems of different complexity are used. A result worth mentioning is due to Schwarz and Truszczyński [140], where they show that skeptical reasoning is the non-monotonic modal logic $S4$ is Π_2^p-complete. This is somehow surprising since reasoning in the *monotonic* modal system $S4$ is PSPACE-complete as proven by Ladner [82]. Marek, Shvarts, and Truszczyński [92] and Gottlob [64] give other interesting complexity results.

As far as first-order languages are concerned, we want to point out that in the original paper by Moore [104] only propositional languages are allowed. By the way, Konolige [79] allows the modal operator to be applied to closed formulae of first-order logic. Under this restriction, Niemelä proves [108] that whenever the satisfiability test in the underlying first-order language is decidable so are all the reasoning tasks of autoepistemic logic.

5. NEGATION IN LOGIC PROGRAMMING

In this section we survey the complexity results for the various semantics for negation in logic programming. This is not intended as an overview of such semantics and their relative merits; a detailed and interesting discussion on this issue can be found in the survey paper by Bidoit [12] and in the paper by Schlipf [138]. We will only marginally refer to the huge body of literature on complexity in rule-based database languages (DATALOG), since this survey focuses on the logic programming semantics more directly related to NMR. For a survey on logic programming and databases we refer the reader to the book by Ceri, Gottlob, and Tanca [30].

We survey the perfect model, stable model, default model, well-founded, supported model, positivistic, inflationary semantics, and the program completion. The expressive power of the various semantics has been studied and compared extensively. For an analysis of this issue see, for example, the work of Schlipf [137]. All these semantics are extensions of the classical minimal model semantics of positive logic programs, and they all agree on the class of stratified and locally stratified logic programs, with exception made for the supported and inflationary semantics and the program completion.

Due to lack of space, this survey will necessarily be incomplete. For a more complete survey on complexity and undecidability results on logic programming we recommend the excellent paper of Schlipf [139].

In the field of logic programming the most important inference task is deciding whether a ground atom belongs to the (usually unique) preferred model of a program. This is the reasoning task we will be referring to in the sequel, where not otherwise specified.

First of all let us recall the main results on the complexity of logic programs without negation. In this case, the preferred model is the unique least Herbrand model. For propositional languages reasoning under such a model can be done in linear time using Dowling and Gallier's algorithm [40]. When dealing with first-order languages, membership of positive and negative literals in the least Herbrand model of a Horn program has been shown to be r.e.-complete and co-r.e.-complete, respectively, by Smullyan [148] and by Andreka and Nemeti [2].

For a first-order language, when negation is allowed there are two different membership problems, depending on which sets of models we are interested in. Traditionally, the semantics of logic programs is given in terms of Herbrand models; however, in some works (e.g., Kunen's paper [81]) it is assumed that models are interpretations over a larger universe containing an infinite number of constants as well as function symbols. As already pointed out by Blair [16], the second alternative leads, in general, to computationally simpler problems. Where not otherwise indicated, we implicitly deal with Herbrand models.

We now analyze the complexity of reasoning tasks in stratified programs. From the computational point of view, these results apply to all the semantics, again with the exception of supported semantics, inflationary semantics, and program completion. In this case it is well known that the unique preferred model of a stratified program can be computed incrementally through the strata by first assigning truth values to the atoms in the lowest stratum and then moving on to the next ones [6]. This holds for locally stratified programs as well. This procedure converges to the iterated least model as defined by Apt, Blair, and Walker [6]. For a propositional language, this immediately implies tractability of the computation, since the computations performed at each step are polynomial and the number of strata is at most linear in the size of the program. Notice also that deciding whether a first-order program is stratified can be done in polynomial time [6], while deciding whether a program is locally stratified is undecidable, as proven by Cholak [33].

The perfect model semantics, introduced by Przymusinski [121], uses the syntactic form of the program to infer a preferential relation on the models. The semantics is not defined for all programs, but for a strict superset of the class of locally stratified programs.

Apt and Blair [5] have proven that if P is a stratified program with n strata, then deciding membership in the perfect model of P is Σ_n^0. Furthermore, for each $n \geq 1$, there is a stratified program P with n strata for which the same problem is Σ_n^0-complete. This result can be specialized to the case of recursion-free programs, in which the problem is r.e. The more general class of locally stratified programs turns out to be computationally more complex. In fact, membership is Δ_1^1-complete over ω, as shown by Cholak and Blair [32] and by Blair, Marek, and Schlipf [17]. A further class of programs that has been analyzed is the class of acyclic programs. For this class of programs Apt and Bezem have proven [4] that deciding membership of ground atoms in the perfect model is a decidable task. The class of acyclic programs is a proper subset of the class of locally stratified ones, and checking acyclicity of a program is a Π_2^0-complete problem.

Using the translations from default logic and circumscription given by Bidoit and Froidevaux [13] and Lifschitz [86], Apt and Blair [5] show that Σ_n^0-hardness is a lower bound for ground inference in restricted classes of default and circumscriptive theories.

For general propositional programs, Eiter and Gottlob prove [42] that deciding whether a perfect model exists is co-NP-hard, while deciding whether a model of a program is perfect is co-NP-complete.

The stable model and default model semantics have been independently introduced by Gelfond and Lifschitz [57] and Bidoit and Froidevaux [13]. While the first is based on autoepistemic logic, the second is based on default logic. As it turned out, these two semantics are equivalent. Since for stratified programs they also coincide with the perfect model semantics, we only consider general programs. Three problems have been analyzed in the propositional case: deciding the existence of a stable model has been proven NP-complete by Marek and Truszczyński [96] and independently by Bidoit and Froidevaux [14]. In [97] Marek and Truszczyński prove that deciding whether an atom is a member of one stable model is NP-complete and membership in all stable models is co-NP-complete.

The supported model semantics has been introduced by Apt, Blair, and Walker [6] to declaratively capture the behavior of PROLOG interpreters. Marek and Truszczyński have shown [96, 97] that deciding the existence of a supported model is an NP-complete problem, membership of an atom in one supported model is also NP-complete, and membership in all supported models is co-NP-complete. A variant of the supported model semantics is the positivistic model semantics introduced by Bidoit and Hull [15]. Schaerf shows [134] that deciding whether a formula is true in one positivistic model is a Σ_2^p-complete problem, while deciding whether a formula is true in all the positivistic models is Π_2^p-complete.

Differently from the previous semantics, the well-founded semantics, introduced by Van Gelder, Ross, and Schlipf [154], is based on three-valued or partial models. Partial models do not necessarily assign the values *true* or *false* to all atoms, but may leave someone undefined. Well-founded semantics always provides a unique preferred model (called well-founded model) to any program; furthermore, if the program is stratified, the well-founded model is completely defined (not partial). The well-founded model of any propositional program can be computed in polynomial time [154].

When a first-order language is used, the computational differences between stable and well-founded model semantics disappear. In fact the problem of deciding membership in all stable (well-founded) models turns out to be Π_1^1-complete over ω for both Herbrand models and models over larger universes, as shown by Schlipf [137, 139] and Van Gelder [153]. Two other problems have been considered by Marek, Nerode, and Remmel. In [91] they show that for the class of programs admitting a unique stable model, the membership problem is still Π_1^1-complete over ω; in [90] they prove that deciding whether a program has a stable model is a Σ_1^1-complete problem over ω. Furthermore, Marek and Subrahmanian show [93] that deciding whether a model is a stable model of a program P is Π_2^0-hard and is in Π_3^0.

As far as the predicate completion defined by Clark [35] is concerned, for propositional languages it has been proven by Kolaitis and Papadimitriou [78] that deciding whether a ground atom belongs to the completion of a program is a co-NP-complete problem. If quantification and function symbols are introduced, then the membership problem over Herbrand models is Π_1^1-complete over ω and over models with larger universes is r.e.-complete.

We do not have the space to refer all the complexity results in the DATALOG area, but we still want to mention the works of Kolaitis and Papadimitriou [78] and Papadimitriou and Yannakakis [114]. In their analysis, Kolaitis and Papadimitriou take into account DATALOG$^{\neg}$ programs, that is, DATALOG programs with negation and with a finite universe, and give two semantics for it. The first is given as the fixpoint of a natural consequence operator. The data-complexity, that is, complexity wrt a fixed extensional database, of finding a fixpoint is NP-complete and finding a least fixpoint is in $F\Delta_2^p$ (for a more precise characterization see [78]). In order to overcome these computational limitations, Kolaitis and Papadimitriou introduce the inflationary semantics, which is based on an iterative construction rather than a fixpoint one. Inflationary semantics, as the well-founded semantics, is defined for all programs and its data-complexity is polynomial.

This analysis has been further developed by Papadimitriou and Yannakakis. The main issue addressed in the paper is to find a semantics whose data complexity is polynomial and which extends the well-founded semantics. The proposed semantics, the tie-breaking semantics, is in some provable sense more general than the well-founded semantics and can be computed in polynomial time. In the same paper they also show that proving whether a fixpoint model exists for any extension of the EDB is undecidable.

An extension of logic programming which has been well analyzed from the computational point of view is disjunctive logic programming, where disjunctions of literals are allowed in the head of the rules. For a comprehensive survey of the semantical and computational aspects of disjunctive logic programming, we refer to the book by Lobo, Minker, and Rajasekar [88]. Due to lack of space we can only briefly mention the major results and give pointers to the literature. Most of the semantics for negation in logic programs have been extended to deal with disjunctive logic programs. In particular, Przymusinski [124] introduces the disjunctive stable model semantics, which extends the stable model semantics, and the partial disjunctive stable model semantics, which extends the well-founded semantics. Eiter and Gottlob in [42] have analyzed the complexity of these and other semantics. In particular they show that deciding whether a literal is true in all the preferred models of a disjunctive logic program under the perfect model semantics and the (partial) disjunctive stable model semantics is a Π_2^p-complete problem. A different extension of the stable model semantics has been given by Gelfond and Lifschitz [58] through the notion of answer set. While inference has been shown to be a problem complete for the second level of the polynomial hierarchy, see Eiter and Gottlob [44], nevertheless some classes have been shown to be computationally simpler by Ben-Eliyahu and Dechter in [11], where they show a polynomial mapping from a subclass of disjunctive logic programs into a propositional theory. Another semantics, called disjunctive database rule, has been defined by Ross and Topor [132] and it has been shown to be polynomially tractable by Chan [31]. Chan has also extended this rule to the possible worlds semantics, which correctly handles negative clauses. However, literal inference under this new semantics is co-NP-complete [31, 42].

6. CIRCUMSCRIPTION AND CLOSED-WORLD REASONING

Circumscription has been defined by McCarthy [98] and refined later by McCarthy himself [99] and by Lifschitz [85]. We refer to the definition which can be found in [99] and [85]. Let $T[P; Z]$ be a first-order formula in which at least the predicates of the disjoint lists of predicates P and Z occur. The *circumscription* $CIRC(T; P; Z)$ *of the predicates* P *in* $T[P; Z]$ *with varying predicates* Z is the following second-order formula:

$$T[P;Z] \wedge (\forall P'Z'.T[P';Z'] \rightarrow \neg(P' < P)), \tag{1}$$

where $P'Z'$ is a list of predicates isomorphic to PZ and $T[P'; Z']$ is T with all the occurrences of predicates of PZ substituted by the corresponding ones in $P'Z'$. The meaning of $P' < P$ is defined in terms of the relation $\leq$. In particular, $P' < P$ is $(P' \leq P) \wedge \neg(P \leq P')$, and $P' \leq P$ stands for the conjunction of the formulae

$$(\forall \vec{x}.p_i'(\vec{x}) \rightarrow p_i(\vec{x}))$$

for each p_i in P. In the above formula p_i p_i' are corresponding predicates, and $\vec{x}$ is an appropriate list of variables. The predicates occurring in P are called *minimized*, while those not occurring in PZ are denoted with Q and are called *fixed*.

The closed-world assumption (CWA) of a first-order formula T is the following formula, defined by Reiter [128]:

$$CWA(T) = T \cup \{\neg p | T \not\models p\}, \tag{2}$$

where p is a ground atom. This rule has been refined by several authors: Minker [102] introduced the *generalized* CWA; Rajasekar, Lobo, and Minker [126], the *weak generalized* CWA; Yahya and Henschen [156], the *extended generalized* CWA; Gelfond and Przymusinska [59] the *careful* CWA; Gelfond, Przymusinsky, and Przymusinska [60], the *extended* CWA and the *iterated* CWA. The notion of fixed and varying predicates has been used in the careful, in the extended and in the iterated CWA. Extended CWA is the most general of all the above rules and is defined as follows:

$$ECWA(T;P;Q;Z) = T \cup \{\neg K \mid \nexists B.(T \not\models B) \wedge (T \models K \vee B)\}, \qquad (3)$$

where $\langle P;Q;Z\rangle$ is a partition of the predicate symbols of T in minimized/fixed/varying ones, K is any formula not involving letters from Z, and B is a disjunction of ground literals whose predicate symbols come from Q and positive ground literals whose predicate symbols come from P. The formulae K whose negations are added to T in the above formula are called *free for negation.*

Relationships between circumscription, closed-world reasoning, and Clark's [35] negation as failure rule have been very well studied in the literature by many researchers, see for example Reiter [130], Sheperdson [146], Lifschitz [84], and in all the works defining closed-world rules. In particular, it has been shown that an abstract notion of minimality underlies definitions (1), (2), and (3). In the propositional case, formulae (1) and (3) are equivalent (see [60]).

6.1. *Inference*

Several computational tasks for circumscription and closed-world reasoning have been addressed from the point of view of the complexity. Inference, that is, to decide whether a first-order formula logically follows from formulae (1) or (3), is the first one we survey.

Circumscription is a second-order formula, and second-order logic is well-known to be computationally strictly harder than first-order logic (see, for example, van Benthem and Doets [152]). In fact circumscription appears to be more expressive than first-order logic, as noticed by Lifschitz. In [84] he shows that second-order formulae like transitive closure, which are not collapsible to first-order ones, are expressible as the circumscription of a first-order formula.

This fact does not *per se* show that circumscription is strictly harder than first-order logic. Nevertheless there have been many attempts at finding classes of first-order formulae whose circumscription is (equivalent to) a first-order formula. From the computational point of view, results of this kind show classes of circumscription in which inference is a r.e. problem.

One of the classes (called of *separable* formulae) has been defined by Lifschitz in [85]. The class of separable formulae is based on a generalization of the idea of predicate completion by Clark [35] and contains all quantifier-free formulae. This result has been improved by Rabinov [125], who showed a class of collapsible formulae that subsumes separable formulae In both cases collapsibility is guaranteed only when three are no varying predicates, that is $Z = \emptyset$.

Kolaitis and Papadimitriou [77] investigated the fragment of existentially quantified formulae, showing that their circumscription is first-order even if varying predicates are allowed.

All the conditions that imply collapsibility are sufficient and not necessary, and the complexity of deciding whether a given formula is in one of the above collapsible cases is in general not known. Actually, the task of proving collapsibility has been proven to be undecidable by Krishnaprasad [80] for general first-order formulae, and by Kolaitis and Papadimitriou for logic programs [77].

All the above results seem to suggest that circumscriptive inference is harder than first-order inference. Actually, Cadoli, Eiter, and Gottlob proved [25] that inference of a formula γ from a circumscription is as hard as testing the validity of any formula in second-order logic.

It is interesting to notice that the last result holds even when γ is propositional, regardless of the presence of varying and/or fixed predicates. The fact that fixed predicates do not contribute to the complexity of circumscription was proved by de Kleer and Konolige [37]. On the other hand the fact that varying predicates do not affect the complexity may seem surprising, in view of the fact that the algorithms for circumscription that have been proposed (for example, Przymusinski's in [122]) are more efficient when varying predicates are not allowed. In the following we will see that the presence of fixed/varying predicates may affect the complexity of inference for some restricted class of formulae.

Other interesting results have been obtained by Schlipf [135], who focused on countably infinite models of formula (1). In particular he shows that, when inference is restricted to such models, then the problem of deciding whether a first-order formula follows from (1) is complete for the class Π_2^1 of the analytical hierarchy over the integers. It is interesting to notice that this result holds even if $Z = \emptyset$.

In [5] Apt and Blair showed a lower bounds of Σ_n^0-hardness for any finite n when ground inference in stratified logic programs is concerned. They take into account a slightly more general version of the formula (1) called *prioritized* circumscription, defined by Lifschitz [85].

As far as the propositional version of (1) is concerned, Eiter and Gottlob showed [41] that the inference problem (with $Q = \emptyset$, $Z = \emptyset$ or not) is complete for the class Π_2^p of the polynomial hierarchy. In [42] they give a more tight result, showing that Π_2^p-completeness holds even if formulae T with no negative clauses are considered. Comparing this result to those in Section 3, we can say that the minimality requirement imposed by circumscription gives a further level of nondeterminism which is analogous to the choice of the right set of defaults. As a consequence a propositional circumscriptive reasoner and a propositional skeptical default reasoner have exactly the same computational power. In Section 8 we see that many researchers attempted at finding analogous relations between the two formalisms.

Papalaskari and Weinstein [115] analyzed the infinitary propositional (sentential) case of circumscription. They showed that when the underlying propositional language is countable and there are neither fixed nor varying predicates, then inference is a problem in Π_2^0 and not in Σ_2^0. It is worth recalling that inference in sentential logic is r.e.

Cadoli and Lenzerini [26] studied the complexity of circumscriptive inference for eight classes of (finite) propositional formulae in which monotonic inference is polynomial. In such a case the upper bound for inference is co-NP.

When the inferred formula γ is a clause, then the problem is co-NP-complete for very restricted cases, for example, when T is both Horn and 2CNF or when T is 2CNF and only positive literals occur in it. The only polynomial case found is when T is Horn, 2CNF, and with no clauses of the form $\neg x \vee \neg y$. Further polynomial cases have been found for restricted versions of the inference problem, for example when T is Horn and $Q = \emptyset$, or when T is 2CNF, γ is a literal, and $Z = \emptyset$.

It is interesting to notice that, while for the general propositional case clause inference is not any harder than literal inference, and fixed/varying letters do not add complexity to inference, this is not the case for restricted cases. As an example, Cadoli and Lenzerini show several cases in which literal inference is polynomial and clause inference is co-NP-complete. A similar tradeoff exists when either $Q = \varnothing$ or $Z = \varnothing$.

One source of complexity of circumscriptive inference seems to be the number of free for negation formulae in (3), which is potentially exponential. Actually, in all the polynomial cases found in [26], it has been shown that only a polynomial number of free for negation formulae has to be taken into account.

Given the equivalence of circumscription and extended closed-world reasoning in the propositional case, the same results hold for the extended CWA as well. For what concerns other forms of closed-world inference, Cadoli and Lenzerini show in the same work that many intractable cases for the extended CWA are tractable for other definitions. As an example, careful closed-world reasoning (defined in [59]) is polynomial for 2CNF formulae. This can be explained by observing that the definition of careful CWA is (3) with K restricted to be a single positive literal. As a consequence the space of the potential free for negation formulae is linear in the size of the input. In view of this fact, it is interesting to notice that Eiter and Gottlob proved [41] that careful closed-world inference is Π_2^p-hard, while the best upper bound is $P^{\Sigma_2^p[O(\log n)]}$, which is worse than the upper bound of Π_2^p that holds for extended closed-world inference.

Other authors have studied the complexity of closed-world inference. Apt and Blair have proven [5] that reasoning under the closed-world assumption (2) is Π_1^0-complete when T is a logic program. Apt [3] had previously shown that the task is polynomial for propositional Horn formulae. The same bounds hold for the weak generalized CWA defined in [126].

Weak generalized CWA has also been studied by Chan [31], who proved that literal inference is polynomial for propositional formulae with no negative clauses. This has also been noticed by Cadoli and Lenzerini [26]. If negative clauses are allowed, then co-NP-hardness holds (see [31]).

As far as the inference under the CWA in an unrestricted propositional language is concerned, Eiter and Gottlob [41] show a lower bound of NP-hardness and an upper bound of $P^{NP[O(\log n)]}$.

In a more recent paper [42], Eiter and Gottlob analyzed iterated CWA, which is a variant of extended CWA defined in [60], showing Π_2^p-completeness results analogous to those obtained for extended CWA.

Chomicki and Subrahmanian analyzed [34] the complexity of generalized closed-world inference for logic programs. The generalized CWA (see [102]) is the rule (3) when $Q = Z = \varnothing$ and K is a positive ground literal. The main result of [34] is that inference of a ground atom under the generalized CWA is complete for the class Π_2^0 of the arithmetic hierarchy.

Borgida and Etherington [18] have shown a very specialized language for representing taxonomies in which some form of generalized closed-world reasoning is polynomial.

The complexity of closed-world reasoning in relational databases has been studied by many authors, for example, Vardi [155]. The problems addressed are outside the scope of this survey, and we refer to the book by Grahne [67] for a general overview.

Several algorithms for inferencing under circumscription have been proposed:

Przymusinski [122], Ginsberg [61], Inoue and Helft [73], Bossu and Siegel [19], Olivetti [110], and Bell et al. [8, 107]. Each of them presupposes languages of limited expressiveness. The algorithms in [122], [73], and [110] use satisfiability testers as subroutines.

6.2. *Satisfiability*

Satisfiability is another computational task that has been addressed. It is well-known that both the circumscription of a first-order formula T and its closure (2) could be unsatisfiable even if T is satisfiable. On the other hand, the extended closure preserves consistency.

Schlipf proved [135] that deciding whether the formula (1) has a countably infinite model is Σ_2^1-complete over the integers.

Eiter and Gottlob [41] proved a lower bound of NP-hardness and an upper bound of $P^{NP[O(\log n)]}$ for consistency checking of the CWA of an arbitrary propositional formula.

These results complement some other by Schlipf, who proved [136] that consistency checking of a slightly different version of the CWA is both NP-hard and co-NP-hard and is in Δ_2^p.

6.3. *Model Checking*

Kolaitis and Papadimitriou [77] noticed that the complexity of circumscription seems to arise even in the problem of *model checking*. Model checking is the problem of deciding whether a finite structure satisfies a given formula Ψ. If Ψ is first-order, then the task is polynomial. On the other hand, in [77] it is shown that when Ψ is the formula (1), then the task is co-NP-complete. The result holds when T is a universal-existential first-order sentence and $Z \neq \emptyset$. co-NP is actually the upper bound for the model checking of the circumscription of all first-order formulae T.

Cadoli [23] strengthens this result in several directions, by proving co-NP-hardness of model checking when the formula T to be circumscribed is propositional, satisfiable, and $Q = Z = \emptyset$; co-NP-hardness in the case $Z = \emptyset$ is implicit in the work of Schlipf [136].

In the same work [23], Cadoli shows several polynomial subcases, for example, when T is a Horn or 2CNF propositional formula. The complexity analysis has been carried out by examining subcases in which Q and/or Z are the empty set. In some interesting cases the task is polynomial if $Q = \emptyset$, and co-NP-complete otherwise. Some polynomial cases are also shown by Eiter and Gottlob [41].

It is interesting to notice that propositional circumscription is in co-NP if model checking is polynomial (see [26]).

Polynomiality of the task when T is a first-order logic program has been shown by Kolaitis and Papadimitriou [77].

6.4. *Model Finding*

Papadimitriou [112] addressed the issue of finding a satisfying truth assignment for the formula (1) when T is propositional. Notice that model finding is a search problem.

The upper bound of the analogous task for plain propositional logic is $F\Delta_2^p$, hence a polynomial number of calls to an NP oracle is sufficient. Gottlob and Fermüller prove [66] that unless P = NP a logarithmic number of calls to an oracle in NP is necessary.

Papadimitriou proves that, if *minimal* satisfying truth assignments are searched, then a polynomial number of calls is still sufficient.

Cadoli [24] addresses the case in which $Q = Z = \emptyset$, proving that the task is $FP^{NP[O(\log n)]}$-hard, even if T is in CNF and a satisfying truth assignment of T is known.

7. ABDUCTION

Abduction is the process of finding explanations for observations in a given theory. This abstract characterization has led to several different definitions of abduction in the literature, which can be roughly divided into two main areas: logic-based abduction as defined by Selman and Levesque [144], and set-covering methods as defined by Reggia, Nau, and Wang in [127]. Although the two methods look formally different, strong connections have been pointed out by Bylander [21] and by Friedrich, Gottlob and Nejdl [53]. In this paper we focus on logic-based abduction, since it has been shown to be tightly related to negation in logic programming (see, for example, Eshghi and Kowalski [49] and Kakas and Mancarella [75]) and default logic (see [144]). Furthermore, we only consider propositional languages, since very little is known about first-order abduction.

Logic-based abduction has been defined as follows [144]: Given a set of propositional clauses Σ and a letter q, an *explanation* for q is a set of literals α such that (1) $\Sigma \cup \alpha \models q$, (2) $\Sigma \cup \alpha$ is consistent, and (3) α is minimal wrt set inclusion. Notice that a trivial explanation satisfying requirements (1) and (3) is q itself. All explanations not containing q will be called *nontrivial*. A condition sometimes imposed on explanations is that the set α must be a subset of a predefined set of literals A which represent the plausible hypotheses. Explanations satisfying this condition are called *assumption-based* explanations in [144].

There are at least three decision tasks which have been considered in the various works on abduction: deciding whether an explanation exists at all, deciding whether an individual hypothesis $h \in A$ belong to at least one acceptable explanation (credulous abductive reasoning), and deciding whether an individual hypothesis $h \in A$ belongs to all the acceptable explanations (skeptical abductive reasoning). The search problem of finding one (best) explanation has also been studied.

The most comprehensive work on the complexity of logic-based abduction has been done by Eiter and Gottlob [47]. In this paper, they analyze the complexity of all the above problems under different preference criteria. In particular they show that when no restrictions are imposed on the syntactic form of the theory Σ and no preference criterion is specified, then all the problems are complete for the second level of the polynomial hierarchy. Adding the requirement of minimality wrt set inclusion does not change the complexity, while using other preference criteria such as minimum cardinality and minimum cardinality with priorities the problems are Δ_3^p-complete. Finally, for minimality wrt set containment with priorities and in the general case of an arbitrary (polynomial) ordering of the explanations, the problems are complete for the third level of the polynomial hierarchy.

This complexity can be reduced of exactly one level if we restrict to theories Σ composed of Horn clauses. A further reduction is sometimes obtained if we restrict to definite Horn clauses.

Under the Horn clause restriction, Selman and Levesque have proven [144] that a nontrivial explanation can be found in polynomial time. They also prove that finding an assumption-based explanation is instead an NP-hard problem, thus showing that the restriction to a subset of the literals may affect the complexity of inference. A similar phenomenon has been noticed for propositional circumscription with and without varying predicates (see Section 6). Bylander [21] has focused on the more restricted case in which Σ is a conjunction of definite clauses, showing a polynomial method for finding assumption-based explanations. Another polynomial subcase has been found by Eshghi [48].

As far as Horn clauses are concerned, Selman and Levesque showed that finding an explanation, with the additional constraint that it has to contain a predefined letter q, is NP-hard.

Friedrich, Gottlob, and Nejdl [53] address problems of skeptical and credulous abductive reasoning in definite Horn clauses, proving that deciding whether a letter belongs to at least one assumption-based explanation is NP-complete, while the set of letters belonging to all the assumption-based explanations can be computed in polynomial time.

The complexity of enumeration problems has also been addressed for abduction. The number of explanations can clearly be exponential wrt the size of Σ, and actually Bylander has proven [21] that determining the number of assumption-based explanations for definite clauses is complete wrt the class #P (see [74]). A related result has been obtained by Friedrich, Gottlob, and Nejdl [53].

Logic-based diagnosis has been proposed as a logical reconstruction of de Kleer's assumption-based truth maintenance systems [36]. Among the computational tasks that have been addressed in this area there is finding explanations, called *nogoods*, of the atom $\perp$ (contradiction). Provan [118, 119] and Rutenburg [133] address the issue of finding the size of a nogood.

In particular, Rutenburg analyzes several definitions of TMS's, considering problems like deciding whether there exists a nogood with size less than a predefined number k. He shows problems whose complexity ranges from polynomial to Σ_2^p-complete. These results confirm the intuition that many computational tasks of non-monotonic reasoning hide two independent sources of complexity whose interaction leads to problems complete for the second level of the polynomial hierarchy.

Other forms of abduction have been presented in the literature, but they are less related to NMR, and for this reason we will not survey them here. However it is worth mentioning the work of Bylander and co-workers [1, 21, 22], who take into account a very general definition of abduction which subsumes that of [144], proving many interesting polynomial as well as NP-hardness results.

8. POLYNOMIAL REDUCTIONS BETWEEN NMR PROBLEMS

Most non-monotonic formalisms have been presented and motivated independently. Nevertheless there have been several attempts at relating them and at finding suitable translations from one logic into another. The existence of these reductions is not only important for a better semantical understanding of the various logics, but it is also useful from the point of view of the complexity analysis.

We focus on polynomial transformations, but recursive nonpolynomial, translations can be used in the same spirit for obtaining new decidability/undecidability results. Actually, every polynomial mapping from one formalism into another can be used for obtaining new complexity results on both formalisms. We would like to illustrate this point by means of a couple of examples, using a well-known translation from circumscriptive into default reasoning proposed by Etherington [50].

Let W be a propositional formula on the alphabet L, and $\langle P; Z\rangle$ a partition of L. Let D be the following set of default rules:

$$D = \{(\ : \ \neg p / \neg p) \mid p \in P\}$$

and Δ the default theory $\langle D, W\rangle$. Etherington [50, Theorem 8.3] shows that inference in $CIRC(W; P; Z)$ and Δ are closely related. Actually, for any formula γ built with the variables of L, $CIRC(W; P; Z) \models \gamma$ if and only if γ follows skeptically from (i.e., in every extension of) Δ. In the jargon of [150], the default rules of D are called *prerequisite-free normal unary* (PFNU).

The above result is clearly a translation that can be computed in polynomial time and proves that skeptical default reasoning is at least computationally as hard as circumscriptive inference, since the latter can be simulated by the former. The translation can be used along with a result by Cadoli and Lenzerini [26] for obtaining a lower bound on the complexity of skeptical default reasoning for a restricted language. In [26] it is shown that when W is 2CNF with no occurrence of negative literals and γ is a literal, then determining whether $CIRC(W; P; Z) \models \gamma$ holds is a co-NP-complete problem. Using Etherington's translation, we have the following new complexity result:

Theorem 1. Let W be a 2CNF formula with no occurrence of negative literals, let D be a set of PFNU default rules, and let l be any literal. Then determining if l is true in all the extensions of the default theory $\langle D, W\rangle$ is a co-NP-hard problem.

Theorem 1 can be interestingly compared to a result by Stillman [150, Theorem 2], stating that if W is a 2CNF formula, D is a set of *prerequisite-free normal* default rules (superset of PFNU), and l is a literal, then the problem of determining whether l is true in at least one extension of $\langle D, W\rangle$ (credulous inference) can be solved in polynomial time. This confirms a previous result by Kautz and Selman, discussed in Section 3, which shows that in languages of limited expressiveness credulous and skeptical default reasoning may have different complexity.

We have just seen how a polynomial reduction can be used for obtaining new intractability results on the complexity of non-monotonic inference. More formally, if we have a polynomial transformation from a decision problem A into a decision problem B and we have a lower bound (e.g., co-NP-hardness) on the complexity of A, then we can prove the same lower bound for B by means of a many–one reduction.

In fact polynomial transformations can be used for obtaining polynomial results as well. For example, the result by Etherington can be read also in the following way: If we had a polynomial algorithm for skeptical reasoning with PFNU defaults, then we could use that algorithm along with the transformation for performing circumscriptive inference in polynomial time.

Actually Kautz and Selman [76] show that, if W is a 1CNF formula, D is a set of *normal unary* default rules (superset of PFNU), and l is a literal, then the problem of determining whether l is true in every extension of $\langle D, W \rangle$ can be solved in polynomial time. Following the previous argument, we have the following result:

Theorem 2. Let W be a 1CNF formula on the alphabet L, let $\langle P; Z \rangle$ be a partition of L, and let l be any literal. The problem of determining of $CIRC(W; P; Z) \models l$ is polynomial.

Although Theorem 2 is implied by a result of Cadoli and Lenzerini [26, Theorem 7], the above construction gives the flavor of the kind of results that can be obtained.

In [27] Cadoli and Lenzerini show a generalization of the reduction by Etherington which allows for fixed predicates. By using this reduction and the complexity results surveyed in Sections 3 and 6, they obtain new complexity results both on default reasoning and on circumscription.

The above analysis shows that polynomial translations between formalisms are very important tools for designers of algorithms for non-monotonic reasoning. This kind of technique has been used for example by Gottlob [64] for proving a lower bound for credulous reasoning in the non-monotonic modal logic N defined by Marek and Truszczyński [95]. Gottlob used the polynomial embedding (defined by Marek and Truszczyński) of default logic into the logic N along with the fact that credulous default reasoning is Σ_2^p-hard, for proving that credulous reasoning in N is Σ_2^p-hard as well.

In the rest of this section we survey some of the polynomial reductions between non-monotonic reasoning problems that have been proposed in the literature. It is worth noticing that those reductions have been studied with the primary goal of comparing semantics and expressiveness and were not intended for complexity analysis. In particular, transformations are usually required to fulfill some abstract criteria. *Modularity*, defined by Imielinski [72] in the context of translations from default logic into circumscription, is one of these criteria. Loosely speaking, a translation is modular if the introduction of new facts does not require recomputation of the whole translation from the beginning.

Grosof [69], Imielinski [72], and Etherington [50, 51] addressed the issue of translating default theories into circumscriptive ones. Transformation for some restricted classes of default theories, such as propositional seminormal defaults without prerequisites, are shown.

Several authors have shown the relations within autoepistemic and default logic. In particular, Konolige [79] has shown a translation from default logic into strongly grounded autoepistemic logic, while Marek and Truszczyński [95] have shown an embedding of default logic in the non-monotonic logic N. In a recent paper [65], Gottlob has shown a nonmodular embedding of default logic into standard autoepistemic logic, also proving that no modular translation is possible. All the above translations are defined for the whole class of propositional default theories. If we restrict our attention to prerequisite-free default theories, then there is a very simple and modular translation which maps single defaults into single modal formulae. This transformation is shown by Gottlob [65] but it is already implicit in the work of Lin and Shoham [87]. We notice that several results surveyed in Section 3 are valid for prerequisite-free default theories.

It is proven by Gelfond and Lifschitz [57] that whenever a program has a single stable model, then the corresponding set of autoepistemic premises has a single stable expansion and the two coincide in the nonmodal part. As a consequence, all the reasoning tasks for the class of autoepistemic theories which are translations of stratified logic programs can be decided in polynomial time. Other reductions from logic programs to default and autoepistemic logic have been shown by Marek and Truszczyński [94].

Cadoli, Eiter, and Gottlob [25] showed a polynomial transformation of any inference problem relative to a circumscription with varying predicates into another inference relative to a circumscription with no varying predicates. Previous attempts at eliminating the varying predicates (see Yuan and Wang [157]) caused exponential growth of the underlying formula. The method in [25] can be used together with another method defined by de Kleer and Konolige [37] for eliminating fixed predicates. Notice, however, that both methods change the syntactic form of the underlying formula, for example, a Horn formula may be changed into a non-Horn one.

9. CONCLUSIONS

In this paper we have surveyed complexity results for many non-monotonic formalisms. In particular, we have focused on those which are more directly related to the field of logic programming. We have seen that all forms of non-monotonicity add a new source of complexity to reasoning. In particular, NMR is strictly harder than classical reasoning for first-order languages and it is harder in the propositional case unless the polynomial hierarchy collapses at a sufficiently low level.

Due to the lack of space we have not been able to report on complexity results for other areas of NMR, such as belief revision, non-monotonic inheritance networks, and the axiomatic approaches to NMR. Nevertheless, we want to give some pointers to the complexity results presented in the literature.

In the area of belief revision and update, important results can be found in the works by Eiter and Gottlob [45, 43], Nebel [105, 106], and Grahne and Mendelzon [68].

The complexity of reasoning in nonmontonic inheritance networks has been analyzed by Horty, Thomason, and Touretzky [71], Selman and Levesque [143], Geffner and Verma [55], Cadoli et al. [28], and Stein [149].

Lehmann and Magidor [83] give some results on the computational complexity of various axiomatic systems for non-monotonic reasoning.

In all the works we have surveyed, tractability is obtained through restricting the expressive power of the language. A rather different approach is based on the use of procedures for *approximated* inference. Some results on the use of approximated inference in NMR can be found in the work by Cadoli and Schaerf [29] and Etherington and Crawford [52].

We are indebted to many persons for their help. We want to thank George Gottlob, Thomas Eiter, Tom Bylander, Rina Dechter, and Rachel Ben-Eliyahu, for finding out mistakes in a previous version and suggesting improvements. Judea Pearl, Jack Minker, Yannis Dimopoulos, Miroslaw Truszczyński, Lynn Andrea Stein, George Gottlob, Thomas Eiter, Rina Dechter, and Rachel Ben-Eliyahu sent us their most recent results. Daniel Lehmann, Henry Kautz, and Krzysztof Apt provided us with more references. Luigia Carlucci Aiello, Andrea Schaerf, Francesco Donini, and Maurizio Lenzerini proof-read earlier versions. The anonymous referees found several mistakes, suggested important improvements in the structure of the paper and noticed many missing references. Their suggestions greatly helped improve the quality of the paper. Furthermore, many thanks are due to all the persons who attended the 4th International Workshop on Non-monotonic Reasoning, held in Vermont in May 1992, who encouraged us in writing this survey paper.

REFERENCES

1. Allemang, D., Tanner, M. C., Bylander, T., and Josephson, J. R., Computational Complexity of Hypothesis Assembly, in: *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87)*, 1987, pp. 1112–1117.
2. Andreka, H., and Nemeti, I., The Generalized Completeness of Horn Predicate Logic as a Programming Language, *Acta Cybernet.*, 4:3–10 (1978).
3. Apt, K. R., Introduction to Logic Programming, in: J. van Leeuwan (ed.), *Handbook of Theoretical Computer Science*, vol. B, Elsevier Science Publishers B. V. (North Holland), 1990, chapter 10.
4. Apt, K. R., and Bezem, M., Acyclic Programs, *New Generation Computing*, 9:335–363 (1991).
5. Apt, K. R. and Blair, H. A., Arithmetic Classification of Perfect Models of Stratified Programs, *Fund. Inform.*, 13:1–18 (1990). [Addendum in 14:339–343 (1991).]
6. Apt, K. R., Blair, H. A., and Walker, A., Towards a Theory of Declarative Knowledge, in: J. Minker (ed.), *Foundation of Deductive Databases and Logic Programming*, Morgan Kaufmann, 1988, pp. 89–142.
7. Baader, F., and Hollunder, B., Embedding Defaults into Terminological Knowledge Representation Formalisms, In: *Proceedings of the Third International Conference on the Principles of Knowledge Representation and Reasoning (KR-92)*, 1992, pp. 306–317.
8. Bell, C., Nerode, A., Ng, R. T., and Subrahmanian, V. S., Implementing Deductive Databases by Linear Programming, in: *Proceedings of the Eleventh Conference on Principles of Database Systems (PODS-92)*, 1992, pp. 283–292.
9. Ben-Eliyahu, R., and Dechter, R., Default Logic, Propositional Logic and Constraints, in: *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, 1991, pp. 379–385.
10. Ben-Eliyahu, R., and Dechter, R., Propositional Semantics for Default Logic, Technical Report R-163, Cognitive Systems Laboratory, Computer Science Department, University of California at Los Angeles, July 1992. [Also presented at the 4th International Workshop on Non-monotonic Reasoning, Vermont, May 1992. Notes edited by H. Kautz and D. W. Etherington.]
11. Ben-Eliyahu, R., and Dechter, R., Propositional Semantics for Disjunctive Logic Programming, in: *Proceedings of the Joint International Conference and Symposium on Logic Programming (JICSLP-92)*, 1992, 379–385. (Full paper to appear in *Ann. Math. and Artificial Intelligence*.)
12. Bidoit, N., Negation in Rule-Based Database Languages: A Survey, *Theoret. Comput. Sci.*, 78:3–83 (1991).
13. Bidoit, N., and Froidevaux, C., General Logic Databases and Programs: Default Logic Semantics and Stratification, *Inform. and Comput.*, 91:15–54 (1991).
14. Bidoit, N., and Froidevaux, C., Negation by Default and Unstratifiable Logic Programs, *Theoret. Comput. Sci.*, 78:85–112 (1991).
15. Bidoit, N., and Hull, R., Minimalism, Justification and Non-monotonicity in Deductive Databases, *J. Comput. System Sci.*, 38:290–325 (1989).
16. Blair, H. A., The Recursion-Theoretic Complexity of the Semantics of Predicate Logic as a Programming Language, *Inform. and Control*, 54:25–47 (1982).

17. Blair, H. A., Marek, W., and Schlipf, J. S., The Expressiveness of Locally Stratified Programs, Technical Report 92-8, Mathematical Sciences Institute, Cornell University, 1992.

18. Borgida, A., and Etherington, D. W., Hierarchical Knowledge Bases and Efficient Disjunctive Reasoning, in: *Proceedings of the First International Conference on the Principles of Knowledge Representation and Reasoning* (*KR-89*), 1989, pp. 33–43.

19. Bossu, P., and Siegel, P., Saturation, Non-monotonic Reasoning and the Closed World Assumption, *Artificial Intelligence*, 25:16–63 (1985).

20. Brachman, R. J. and Levesque, H. J., A Fundamental Tradeoff in Knowledge Representation and Reasoning, in: R. J. Brachman and H. J. Levesque (eds.), *Readings in Knowledge Representation*, Morgan Kaufmann, 1985, pp. 41–70.

21. Bylander, T., The Monotonic Abduction Problem: A Functional Characterization on the Edge of Tractability, in: *Proceedings of the Second International Conference on the Principles of Knowledge Representation and Reasoning* (*KR-91*), 1991, pp. 70–77.

22. Bylander, T., Allemang, D., Tanner, M. C., and Josephson, J. R., The Computational Complexity of Abduction, *Artificial Intelligence*, 49:25–60 (1991).

23. Cadoli, M., The Complexity of Model Checking for Circumscriptive Formulae, *Inform. Process. Lett.*, 44:113–118 (1992).

24. Cadoli, M., On the Complexity of Model Finding for Non-monotonic Propositional Logics, in: A Marchetti Spaccamela, P. Mentrasti, and M. Venturini Zilli (eds.), *Proceedings of the Fourth Italian Conference on Theoretical Computer Science*, World Scientific, 1992, pp. 125–139.

25. Cadoli, M., Eiter, T., and Gottlob, G., An Efficient Method for Eliminating Varying Predicates from a Circumscription, *Artificial Intelligence*, 54:397–410 (1992).

26. Cadoli, M., and Lenzerini, M., The Complexity of Closed World Reasoning and Circumscription, in: *Proceedings of the Eighth National Conference on Artificial Intelligence* (*AAAI-90*), MIT Press, 1990, pp. 550–555. (Extended version appears as Technical Report RAP 17.91, Università di Roma "La Sapienza," Dipartimento di Informatica e Sistemistica, 1991. To appear in *J. Comput. System Sci.*)

27. Cadoli, M., and Lenzerini, M., Mapping Complexity Results from Circumscription to Default Logic and Vice Versa, unpublished manuscript, July 1991.

28. Cadoli, M., Lenzerini, M., Nardi, D., and Pirri, F., Circumscription and Non-monotonic Inheritance, in: *Proceedings of the 1st Pacific Rim Conference on Artificial Intelligence* (*PRICAI-90*), 1990, pp. 760–765.

29. Cadoli, M., and Schaerf, M., Approximate Inference in Default Reasoning and Circumscription, in: B. Neumann (ed.), *Proceedings of the Tenth European Conference on Artificial Intelligence* (*ECAI-92*), Wiley, 1992, pp. 319–323. (Extended version presented at the 4th International Workshop on Non-monotonic Reasoning, Vermont, May 1992. Notes edited by H. Kautz and D. Etherington. To appear in *Fund. Inform.*)

30. Ceri, S., Gottlob, G., and Tanca, L., *Logic Programming and Databases*, Springer-Verlag, Berlin, 1990.

31. Chan, E., A Possible Worlds Semantics for Disjunctive Databases, *IEEE Trans. Data and Knowledge Engrg.*, 1993, to appear.

32. Cholak, C., and Blair, H., The Complexity of Local Stratification, Technical Report SU-CIS-91-27, School of Computer and Information Science, Syracuse University, July 1991.

33. Cholak, P., Post Correspondence Problem and Prolog Programs, Technical Report, University of Wisconsin, Madison, 1988.

34. Chomicki, J., and Subrahmanian, V. S., Generalized Closed World Assumption is Π_2^0-Complete, *Inform. Process. Lett.*, 34:289–291 (1990).

35. Clark, K. L., Negation as Failure, in: H. Gallaire and J. Minker (eds.), *Logic and Data Bases*, Plenum, 1978, pp. 293–322.

36. de Kleer, J., An Assumption-Based TMS, *Artificial Intelligence*, 28:127–162 (1986).

37. de Kleer, J., and Konolige, K., Eliminating the Fixed Predicates from a Circumscription, *Artificial Intelligence*, 39:391–398 (1989).

38. Dimopoulos, Y., and Magirou, V., A Graph Theoretic Approach to Default Logic, *Inform. and Comput.*, 1993, to appear.

39. Donini, F. M., Lenzerini, M., Nardi, D., Pirri, F., and Schaerf, M., Non-monotonic Reasoning, *Artificial Intelligence Rev.*, 4:163–210 (1990).

40. Dowling, W. P., and Gallier, J. H., Linear-Time Algorithms for Testing the Satisfiability of Propositional Horn Formulae, *J. Logic Programming*, 1:267–284 (1984).

41. Eiter, T., and Gottlob, G., Propositional Circumscription and Extended Closed World Reasoning are Π_2^P-complete, *Theoret. Comput. Sci.*, 114:231–245 (1993).

42. Eiter, T., and Gottlob, G., Complexity Aspects of Various Semantics for Disjunctive Databases, in: *Proceedings of the Twelfth Conference on Principles of Database Systems (PODS-93)*, 1993, pp. 158–167.

43. Eiter, T., and Gottlob, G., The Complexity of Nested Counterfactuals and Iterated Knowledge Base Revision, in: *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, 1993, in press.

44. Eiter, T., and Gottlob, G., Complexity Results for Disjunctive Logic Programming and Application to Non-monotonic Logics, Technical Report CD-TR 92/41, Technische Universität Wien, Vienna, Austria, Christian Doppler Labor für Expertensysteme, October 1992.

45. Eiter, T., and Gottlob, G., On the Complexity of Propositional Knowledge Base Revision, Updates and Counterfactuals, *Artificial Intelligence*, 57:227–270 (1992).

46. Eiter, T., and Gottlob, G., Reasoning with Parsimonious and Moderately Grounded Expansions, *Fund. Inform.*, 17:31–54 (1992).

47. Eiter, T., and Gottlob, G., The Complexity of Logic-Based Abduction, in: *Tenth Symposium on Theoretical Aspects of Computer Science* (*STACS-93*), 1993. (Extended version in press as Technical Report CD-TR 92/35 Technische Universität Wien, Vienna, Austria, Christian Doppler Labor für Expertensysteme.)

48. Eshghi, K., A Tractable Class of Abduction Problems, in: *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, 1993, in press.

49. Eshghi, K., and Kowalski, R. A., Abduction Compared with Negation by Failure, in: *Proceedings of the Sixth International Conference on Logic Programming* (*ICLP-89*), 1989, pp. 234–254.

50. Etherington, D. V., *Reasoning with Incomplete Information*, Morgan Kaufman, Los Altos, CA, 1987.

51. Etherington, D. V., Relating Default Logic and Circumscription, in: *Proceedings of the Tenth International Joint Conference on Artificial Intelligence* (*IJCAI-87*), 1987, pp. 489–494.

52. Etherington, D. V., and Crawford, J. M., Towards Efficient Default Reasoning, presented at the 4th International Workshop on Non-monotonic Reasoning, Vermont, May 1992. (Notes edited by H. Kautz and D. Etherington, 1992.)

53. Friedrich, G., Gottlob, G., and Nejdl, W., Hypothesis Classification, Abductive Diagnosis and Therapy, in: *Proceedings of the International Workshop on Expert Systems in Engineering, Lecture Notes In Artificial Intelligence*, 462, Springer-Verlag, 1990, pp. 69–78.

54. Garey, M., and Johnson, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.

55. Geffner, H., and Verma, T., Inheritance = Chaining + Defeat, in: *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems* (*ISMIS-89*), North Holland, 1989, pp. 411–418.

56. Gelfond, M., On Stratified Autoepistemic Theories, in: *Proceedings of the Sixth National Conference on Artificial Intelligence* (*AAAI-87*), Morgan Kaufmann, 1987, pp. 207–211.

57. Gelfond, M., and Lifschitz, V., The Stable Model Semantics for Logic Programming, in: *Proceedings of the Fifth Logic Programming Symposium*, MIT Press, 1988, pp. 1070–1080.

58. Gelfond, M., and Lifschitz, V., Classical Negation in Logic Programs and Disjunctive Databases, *New Generation Computing*, 9:365–385 (1991).

59. Gelfond, M., and Przymusinska, H., Negation as Failure: Careful Closure Procedure, *Artificial Intelligence*, 30:273–287 (1986).

60. Gelfond, M., and Przymusinska, H., and Przymusinsky, T., On the Relationship between Circumscription and Negation as Failure, *Artificial Intelligence*, 38:49–73 (1989).

61. Ginsberg, M. L., A Circumscriptive Theorem Prover, *Artificial Intelligence*, 39:209–230 (1989).

62. Goldszmidt, M., and Pearl, J., Rank-Based Systems: A Simple Approach to Belief Revision, Belief Update and Reasoning about Evidence and Actions, in *Proceedings of the Third International Conference on the Principles of Knowledge Representation and Reasoning* (*KR-92*), 1992, pp. 661–672.

63. Gottlob, G., The Complexity of Propositional Default Reasoning under the Stationary Fixed Point Semantics, Technical Report CD-TR 92/42, Technische Universität Wien, Vienna, Austria, Christian Doppler Labor für Expertensysteme, 1992.

64. Gottlob, G., Complexity Results for Non-monotonic Logics, *J. Logic and Comput.*, 2:397–425 (1992).

65. Gottlob, G., On the Power of Pure Beliefs or Embedding Default Logic into Standard Autoepistemic Logic, in: *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, 1993, in press.

66. Gottlob, G., and Fermüller, C. G., Removing Redundancy from a Clause, *Artificial Intelligence Journal*, 61:263–289, (1993).

67. Grahne, G., *The Problem of Incomplete Information in Relational Databases*, *Lecture Notes in Computer Science*, 554, Springer-Verlag, 1991.

68. Grahne, G., and Mendelzon, A. O., Updates and Subjunctive Queries, presented at the Workshop on Nonstandard queries and answers, Toulouse, France, July 1991. (Notes edited by R. Demolombe, L. Fariñas del Cerro, and T. Imielinski, 1991.)

69. Grosof, B., Default Reasoning as Circumscription, presented at the First International Workshop on Non-monotonic Reasoning, 1984.

70. Hinman, P. G., *Recursion-Theoretic Hierarchies*, Springer-Verlag, 1978.

71. Horty, J. F., Thomason, R. H., and Touretzky, D. S., A Skeptical Theory of Inheritance in Non-monotonic Semantic Networks, *Artificial Intelligence*, 42:311–348 (1990).

72. Imielinski, T., Results on Translating Defaults to Circumscription, *Artificial Intelligence*, 32:131–146 (1987).

73. Inoue, K., and Helft, N., On Theorem Provers for Circumscription, in: *Proceedings of the Eighth Biennal Conference of the Canadian Society for Computational Studies of Intelligence* (*CSCSI '90*), Morgan Kaufmann, 1990.

74. Johnson, D. S., A Catalog of Complexity Classes, in: J. van Leeuwen (ed.), *Handbook of Theoretical Computer Science*, vol. A, Elsevier Science Publishers B. V. (North Holland), 1990, chapter 2.

75. Kakas, A. C., and Mancarella, P., Negation as Stable Hypothesis, in: *Proceedings of the First International Workshop on Logic Programming and Non Monotonic Reasoning*, MIT Press, 1991, pp. 275–288.

76. Kautz, H. A., and Selman, B., Hard Problems for Simple Default Logics, *Artificial Intelligence*, 49:243–279 (1991).

77. Kolaitis, P. G., and Papadimitriou, C. H., Some Computational Aspects of Circumscription, *J. ACM*, 37:1–14 (1990).

78. Kolaitis, P. G., and Papadimitriou, C. H., Why Not Negation by Fixpoint?, *J. Comput. System Sci.*, 43:125–144 (1991).

79. Konolige, K., On the Relationship between Default and Autoepistemic Logic, *Artificial Intelligence*, 35:343–382 (1988).
80. Krishnaprasad, T., On the Computability of Circumscription, *Inform. Process. Lett.*, 27:237–243 (1988).
81. Kunen, K., Negation in Logic Programming, *J. Logic Programming*, 4:289–308 (1987).
82. Ladner, R. E., The Computational Complexity of Provability in Systems of Modal Propositional Logic, *SIAM J. Comput.*, 6:467–480 (1977).
83. Lehmann, D., and Magidor, M., What Does a Conditional Knowledge Base Entail?, *Artificial Intelligence*, 55:1–60, (1992).
84. Lifschitz, V., Closed-World Databases and Circumscription, *Artificial Intelligence*, 27:229–235 (1985).
85. Lifschitz, V., Computing Circumscription, in: *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (*IJCAI-85*), 1985, pp. 121–127.
86. Lifschitz, V., On the Declarative Semantics of Logic Programs with negation, in: J. Minker (ed.), *Foundation of Deductive Databases and Logic Programming*, Morgan Kaufmann, 1988, pp. 177–192.
87. Lin, F., and Shoham, Y., A Logic of Knowledge and Justified Assumptions, *Artificial Intelligence*, 57:271–290 (1992).
88. Lobo, J., Minker, J., and Rajasekar, *Foundations of Disjunctive Logic Programming*, MIT Press, Cambridge, MA, 1992.
89. Lukaszewicz, W., *Non-Monotonic Reasoning*, Ellis-Horwood, Chichester, 1990.
90. Marek, W., Nerode, A., and Remmel, J., The Stable Models of a Predicate Logic Program, in: *Joint International Conference and Symposium on Logic Programming*, 1992, pp. 446–460.
91. Marek, W., Nerode, A., and Remmel, J., A Theory of Nonmonotonic Rule Systems II. *Ann. of Math. and Artificial Intelligence*, 5:229–264, 1992.
92. Marek, W., Shvarts, G. F., and Truszczyński, M., Model Non-monotonic Logics: Ranges, Characterization, Computation, in: *Proceedings of the Second International Conference on the Principles of Knowledge Representation and Reasoning* (*KR-91*), 1991, pp. 395–404.
93. Marek, W., and Subrahmanian, V. S., The Relationship between Logic Program Semantics and Non-monotonic Reasoning, in: *Proceedings of the Sixth International Conference on Logic Programming* (*ICLP-89*), 1989, pp. 600–617.
94. Marek, W., and Truszczyński, M., Stable Semantics for Logic Programs and Default Theories, in: E. Lusk and R. Overbeek (eds.), *Proceedings of the North American Conference on Logic Programming*, MIT Press, 1989, pp. 243–256.
95. Marek, W., and Truszczyński, M., Modal Logic for Default Reasoning, *Ann. of Math. and Artificial Intelligence*, 1:275–302 (1990).
96. Marek, W., and Truszczyński, M., Autoepistemic Logic, *J. ACM*, 38:588–619 (1991).
97. Marek, W., and Truszczyński, M., Computing Intersection of Autoepistemic Expansions, in: *Proceedings of the First International Workshop on Logic Programming and Non Monotonic Reasoning*, MIT Press, 1991, pp. 37–50.
98. McCarthy, J., Circumscription—A Form of Non-Monotonic Reasoning, *Artificial Intelligence*, 13:27–39 (1980).
99. McCarthy, J., Applications of Circumscription to Formalizing Commonsense knowledge, *Artificial Intelligence*, 28:89–116 (1986).
100. McDermott, D., Non-monotonic Logic II: Mon-monotonic Modal Theories, *J. ACM*, 29:33–57 (1982).
101. McDermott, D., and Doyle, J., Non-monotonic Logic I, *Artificial Intelligence*, 13:41–72 (1980).
102. Minker, J., On Indefinite Databases and the Closed World Assumption, in: *Proceedings of the Sixth Conference on Automated Deduction* (*CADE-82*), 1982, pp. 292–308.
103. Minker, J., An Overview of Non-monotonic Reasoning and Logic Programming, Technical Report CS-TR-2736, Department of Computer Science, University of Maryland, August 1991. (Invited banquet address, First International Workshop on LP and NMR, 1991.)

104. Moore, R. C., Semantical Considerations on Non-monotonic Logic, *Artificial Intelligence*, 25:75–94 (1985).
105. Nebel, B., A Knowledge-Level Analysis of Belief Revision, in: *Proceedings of the First International Conference on the Principles of Knowledge Representation and Reasoning* (*KR-89*), 1989, pp. 301–311.
106. Nebel, B., Belief Revision and Default Reasoning: Syntax-Based Approaches, in: *Proceedings of the Second International Conference on the Principles of Knowledge Representation and Reasoning* (*KR-91*), 1991, pp. 417–428.
107. Nerode, A., Ng, R. T., and Subrahmanian, V. S., Computing Circumscriptive Databases, Part I: Theory and Algorithms, Technical Report CS-TR-2824, University of Maryland, Department of Computer Science, 1992.
108. Niemelä, I., On the Decidability and Complexity of Autoepistemic Reasoning, *Fund. Inform.*, 17:117–156 (1992).
109. Niemelä, I., and Rintanen, J., On the Impact of Stratification on the Complexity of Non-monotonic Reasoning, in: *Proceedings of the Third International Conference on the Principles of Knowledge Representation and Reasoning* (*KR-92*), 1992, pp. 627–638.
110. Olivetti, N., Tableaux and Sequent Calculus for Minimal Entailment, *J. Automated Reasoning*, 9:99–139 (1992).
111. Papadimitriou, C. H., On the Complexity of Unique Solutions, *J. ACM*, 31:392–400 (1984).
112. Papadimitriou, C. H., On Selecting a Satisfying Truth Assignment (Extended Abstract), in: *Proceedings of the 32nd Annual Symposium on the Foundations of Computer Science* (*FOCS-91*), 1991, pp. 163–169.
113. Papadimitriou, C. H., and Sideri, M., On Finding Extensions of Default Theories, in: *Proceedings of the International Conferences on Database Theory* (*ICDT-92*), 1992, pp. 276–281.
114. Papadimitriou, C. H., and Yannakakis, M., Tie-Breaking Semantics and Structural Totality, in: *Proceedings of the Eleventh Conference on Principles Of Database Systems* (*PODS-92*), 1992, pp. 16–22.
115. Papalaskari, M., and Weinstein, S., Minimal Consequence in Sentential Logic, *J. Logic Programming*, 9:19–31 (1990).
116. Pearl, J., Probabilistic Semantics for Non-monotonic Reasoning: A Survey, in: R. Cummins and J. Pollock (eds.), *Philosophy and AI: Essays at the Interface*, Bradford Books/MIT Press, Cambridge, MA, 1991, pp. 157–187.
117. Poole, D. L., Goebel, R. G., and Aleliunas, R., Theorist: A Logical Reasoning System for Defaults and Diagnosis, in: N. Cercone and G. McCalla (eds.), *The Knowledge Frontier: Essays in the Representation of Knowledge*, Springer, New York, 1987, pp. 331–352.
118. Provan, G. M., A Complexity Analysis of Assumption-Based Truth Maintenance Systems, in: B. Smith and G. Kelleher (eds.), *Reason Maintenance Systems and their Applications*, Ellis Horwood, New York, 1988, pp. 98–113.
119. Provan, G. M., The Computational Complexity of Multiple-Context Truth Maintenance Systems, in: *Proceedings of the Ninth European Conference on Artificial Intelligence* (*ECAI-90*), 1990, pp. 522–527.
120. Przymusinska, H., and Przymusinsky, T., Stationary Default Extensions, presented at the 4th International Workshop on Non-monotonic Reasoning, Vermont, May 1992. (Notes edited by H. Kautz and D. Etherington, 1992.)
121. Przymusinski, T., On the Declarative Semantics of Stratified Deductive Databases and Logic Programs, in: J. Minker (ed.), *Foundation of Deductive Databases and Logic Programming*, Morgan Kaufmann, 1988, pp. 193–216.
122. Przymusinski, T., An Algorithm to Compute Circumscription, *Artificial Intelligence*, 38:49–73 (1989).
123. Przymusinski, T., Non-monotonic Formalisms and Logic Programming, in: *Proceedings of the Sixth International Conference of Logic Programming*, MIT Press, 1989, pp. 655–674.

124. Przymusinski, T., Stable Semantics for Disjunctive Programs, *New Generation Computing*, 9:401–424 (1991).
125. Rabinov, A., A Generalization of Collapsible Cases of Circumscription, *Artificial Intelligence*, 38:111–117 (1989).
126. Rajasekar, A., Lobo, J., and Minker, J., Weak Generalized Closed World Assumption, *J. Automated Reasoning*, 5:293–307 (1989).
127. Reggia, J. A., Nau, D. S., and Wang, P., Diagnostic Expert Systems Based on a Set Covering Model, *Int. J. Man-Machine Studies*, 19:437–460 (1983).
128. Reiter, R., On Closed World Data Bases, in: H. Gallaire and J. Minker (eds.), *Logic and Data Bases*, Plenum, 1978, pp. 119–140.
129. Reiter, R., A Logic for Default Reasoning, *Artificial Intelligence*, 13:81–132 (1980).
130. Reiter, R., Circumscription Implies Predicate Completion (Sometimes), in: *Proceedings of the National Conference on Artificial Intelligence* (*AAAI-92*), 1982, pp. 418–420.
131. Rogers, H., *Theory of Recursive Functions and Effective Computability*, McGraw-Hill, 1967.
132. Ross, K., and Topor, R., Inferring Negative Information from a Disjunctive Database, *J. Automated Reasoning*, 4:397–424 (1988).
133. Rutenburg, V., Computational Complexity of Truth Maintenance Systems, in: *Eighth Symposium on Theoretical Aspects of Computer Science* (*STACS-91*), *Lecture Notes in Computer Science*, 480, Springer-Verlag, 1991.
134. Schaerf, M., Logic Programming and Autoepistemic Logics: New Relations and Complexity Results, in: *Proceedings of the Third Congress of the Italian Association for Artificial Intelligence (AI*AI-93)*, Lecture Notes in Artificial Intelligence, Springer-Verlag, 1993, in press.
135. Schlipf, J. S., Decidability and Definability with Circumscription, *Ann. Pure Appl. Logic*, 35:173–191 (1987).
136. Schlipf, J. S., When Is Closed World Reasoning Tractable?, in: *Proceedings of the Third International Symposium on Methodologies for Intelligent Systems* (*ISMIS-88*), North Holland, 1988, pp. 485–494.
137. Schlipf, J. S., The Expressive Powers of the Logic Programming Semantics, in: *Proceedings of the Ninth Conference on Principles of Database Systems* (*PODS-90*), 1990, pp. 196–204. (Expanded version available as University of Cincinnati Computer Science Department Technical Report CIS-TR-90-3.)
138. Schlipf, J. S., Formalizing a Logic for Logic Programming, *Ann. Math. and Artificial Intelligence*, 5:279–302 (1992).
139. Schlipf, J. S., A survey of Complexity and Undecidability Results in Logic Programming, in: *Proceedings of the Workshop on Structural Complexity and Recursion-Theoretic Methods in Logic Programming*, Washington, DC, 1992, pp. 143–164.
140. Schwarz, G., and Truszczyński, M., Non-monotonic Reasoning Is Sometimes Easier, Technical Report 213-92, Department of Computer Science, University of Kentucky, 1992.
141. Schwind, C. B., A Tableaux-Based Theorem Prover for a Decidable Subset of Default Logic, in: *Proceedings of the Tenth Conference on Automated Deduction* (*CADE-90*), 1990, pp. 528–542.
142. Selman, B., and Kautz, H. A., Model-Preference Default Theories, *Artificial Intelligence*, 45:287–322 (1990).
143. Selman, B., and Levesque, H. J., The Tractability of Path-Based Inheritance, in: *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (*IJCAI-89*), 1989, pp. 1140–1145.
144. Selman, B., and Levesque, H. J., Abductive and Default Reasoning: A Computational Core, in: *Proceedings of the Eighth National Conference on Artificial Intelligence* (*AAAI-90*), 1990, pp. 343–348.
145. Selman, B., Levesque, H. J., and Mitchell, D., A New Method for Solving Hard Satisfiability Problems, in: *Proceedings of the Tenth National Conference on Artificial Intelligence* (*AAAI-92*), 1992, pp. 440–446.

146. Sheperdson, J. C., Negation as Failure: A Comparison of Clark's Completed Data Base and Reiter's Closed-World Assumption, *J. Logic Programming*, 1:51–79 (1984).
147. Shvarts, G., Autoepistemic Modal Logics, in: *Proceedings of the Third Conference on Theoretical Aspects of Reasoning about Knowledge (TARK-90)*, pp. 97–109.
148. Smullyan, R. M., *First Order Logic*, Springer-Verlag, Berlin, 1968.
149. Stein, L. A., Resolving Ambiguity in Non-monotonic Inheritance Hierarchies, *Artificial Intelligence*, 55:259–310 (1992).
150. Stillman, J., It's Not My Default: The Complexity of Membership Problems in Restricted Propositional Default Logics, in: *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, 1990, pp. 571–578.
151. Stillman, J., The Complexity of Propositional Default Logics, in: *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, 1992, pp. 794–799.
152. van Benthem, J., and Doets, K., Higher Order Logic, in: D. Gabbay and F. Guenthner (eds.), *Handbook of Philosophical Logic*, 1, D. Reidel, Dordrecht, 1983, chapter 4.
153. Van Gelder, A., The Alternating Fixpoint of Logic Programs with Negation, in: *Proceedings of the Eighth Conference on Principles of Database Systems (PODS-89)*, 1989, pp. 1–10.
154. Van Gelder, A., Ross, K. A., and Schlipf, J. S., The Well-Founded Semantics for General Logic Programs, *J. ACM*, 38:620–650 (1991).
155. Vardi, M. Y., Querying Logical Databases, *J. Comput. System Sci.*, 33:142–160 (1986).
156. Yahya, A., and Henschen, L. J., Deduction in Non-Horn Databases, *J. Automated Reasoning*, 1:141–160 (1985).
157. Yuan, L. Y., and Wang, C. H., On Reducing Parallel Circumscription, in: *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, 1988, pp. 460–464.
158. Zhang, A., and Marek, W., On the Classification and Existence of Structures in Default Logic, *Fund. Inform.*, 13:485–499 (1990).