Proof Systems for Nonmonotonic Logics

Ilkka Niemelä
Department of Computer Science
Helsinki University of Technology
Otakaari 1, FIN–02150 Espoo
Finland
Ilkka.Niemela@hut.fi

Camilla Schwind
Laboratoire d'Informatique de Marseille CNRS
Faculté des Sciences de Luminy
163, Avenue de Luminy, Case 901
13288 Marseille, Cedex 9, France
schwind@lim.univ-mrs.fr

Abstract

A survey of proof systems for nonmonotonic logics is presented. In particular, proof methods for default logics, nonmonotonic modal logics, circumscription, and conditional logics are discussed.

1 Introduction

Human commonsense reasoning and reasoning in artificial intelligence programs typically violates one of the characteristic features of logical reasoning: the monotonicity property.

> A logical calculus is said to be *monotonic* if its derivability relation $\vdash$ has the following property: if $\Sigma \vdash \phi$ and $\Sigma \subseteq \Sigma'$, then $\Sigma' \vdash \phi$ for any sets of formulae Σ and Σ' and a formula ϕ.

Reasoning not satisfying the monotonicity property is said to be *nonmonotonic*. People "jump" to conclusions which they may later retract when given more information. The conclusion that the local cafeteria is open at 2 p.m. on Friday is a typical example of a conclusion that would be retracted when given more information, e.g. that the cafeteria is under repair. Artificial intelligence programs must also be able to reason nonmonotonically. For instance, databases where the *closed world assumption* [101] is used, logic programming where the *negation as failure* rule [17] is applied to implement a form of negation, inheritance hierarchies where properties are *inherited by default* [83], reasoning about action where the *frame problem* [80] and the *qualification problem* [77] have to be solved as well as *diagnosis* [104] are all examples of areas where reasoning patterns not satisfying the monotonicity property are used. As the list illustrates, nonmonotonic reasoning is one of fundamental phenomena in knowledge representation and reasoning. However, the exact nature of nonmonotonic reasoning is somewhat difficult to capture. The following example demonstrates the difficulties.

Example 1 Consider the following pieces of knowledge.

- A quaker who is not known to be a hawk is a pacifist.
- A republican who is not known to be a pacifist is a hawk.

We use a PROLOG system to represent the knowledge and interpret "not known" as PROLOG's negation (as failure). For simplicity we consider a given person so that no variables are needed. If we assume that the person is both a quaker and a republican, the resulting set of PROLOG rules is given below.

```
pacifist:- quaker, not(hawk).
hawk:- republican, not(pacifist).
quaker.
republican.
```

Given a query `pacifist`, a standard PROLOG system loops forever. But what is the right answer to the query? Usually in a knowledge representation system, answers are defined by giving a procedure which computes the answers by examining the knowledge base containing some symbolic representation of the pieces of knowledge. Assume that an answer to the query `pacifist` is provided by such a system. The answer is a function of the method of breaking the recursion in the two rules above and thus it is tightly coupled with a particular implementation of the procedure computing the answers. This kind of a situation makes concepts like negation as failure employed in knowledge representation difficult to understand and use. Maintenance and updating of such a system becomes tedious. Comparison of different systems is seriously hindered. Correctness of such a system is harder to assess. Areas of optimization are limited. A declarative way of characterizing correct answers would improve the situation considerably. By such a characterization we mean a way of defining correct answers without any reference to a particular procedure. But as the reasoning patterns defy one of the fundamental properties of classical logical reasoning, such a characterization is not straightforward to devise. In the case of logic programs this has resulted in a large number of competing proposals. For example, Przymusinska and Przymusinski [95] and Bidoit [9] offer surveys of the various approaches. ■

An obvious approach to characterizing nonmonotonic reasoning declaratively is to devise a suitable logical calculus that captures the correct reasoning patterns. A large number of researchers have studied the formal underpinnings of nonmonotonic reasoning and several *nonmonotonic logics* have been proposed.

The best symbolic approaches to formalizing nonmonotonic reasoning developed in the seventies were collected in a special issue of *Artificial Intelligence* (volume 13, 1980). Ginsberg has compiled a number of the most prominent papers on nonmonotonic reasoning which appeared before 1987 in a single edition [35]. From this work, *default logic* by Reiter [102], the family of circumscription type formalizations [78, 79], McDermott and Doyle style *nonmonotonic modal logics* [82, 81],

and *autoepistemic logic* [84] stand out as most popular logical formalizations of nonmonotonic reasoning. For example, the article of Reiter [103] and the books of Etherington [25], Besnard [7], and Brewka [13] provide introductions to nonmonotonic reasoning.

In this article we first review briefly the basic notions of default logic, nonmonotonic modal logics, and circumscription, present key complexity results and discuss relationships of between the approaches. Then we cover in detail proofs systems for default logics, nonmonotonic modal logics, circumscription, and conditionals.

2 Formalisms for Nonmonotonic Reasoning

2.1 Default Logic

Reiter's [102] default logic is one of the leading formalizations of nonmonotonic reasoning. It attempts to formalize default reasoning of the form "in the absence of any information to the contrary, assume ...". This is achieved by extending classical predicate logic with *default rules* of the form

$$\frac{\alpha : \beta_1, \ldots, \beta_n}{\gamma} \tag{1}$$

where $\alpha, \beta_1, \ldots, \beta_n, \gamma$ are all closed formulae of predicate logic. The rule can be read as if the *prerequisite* α holds and it is consistent to assume all the *justifications* $\beta_1, \ldots, \beta_n$, then the *consequent* γ holds.

Statements of the type "birds typically fly" are formalized in default logic using a default rule stating that a bird for which it is consistent to assume that it flies also flies:

$$\frac{b : f}{f}.$$

In default logic knowledge is represented by a pair (D, W) where D is a set of default rules and W is a set of ordinary first-order sentences. A possible set of correct nonmonotonic inferences drawn from a *default theory* (D, W) is called an *extension* of (D, W). When presenting Reiter's definition of an extension, we use an operator $\text{apply}_{D,E}(E')$ which applies the default rules from D whose prerequisites are known (are in E') and whose justifications are consistent with E.

$$\begin{aligned} &\text{apply}_{D,E}(E') = \\ &\quad \{\gamma \mid \frac{\alpha : \beta_1, \ldots, \beta_n}{\gamma} \in D, \alpha \in E', \neg\beta_1 \notin E, \ldots, \neg\beta_n \notin E\} \end{aligned} \tag{2}$$

Definition 2.1 (Reiter [102]) *Let (D, W) be a default theory. For any set of first-order sentences E, let $\Gamma_{(D,W)}(E)$ be the smallest set satisfying the following properties*

1. $W \subseteq \Gamma_{(D,W)}(E)$,

2. $\mathrm{Cn}(\Gamma_{(D,W)}(E)) = \Gamma_{(D,W)}(E)$

3. $\mathrm{apply}_{D,E}(\Gamma_{(D,W)}(E)) \subseteq \Gamma_{(D,W)}(E)$

where $\mathrm{Cn}(\Sigma) = \{\phi \mid \Sigma \models \phi\}$ *and* $\models$ *denotes the first-order consequence relation. A set* $E \subseteq \mathcal{L}$ *is an* extension *of the default theory* (D, W) *if and only if*

$$E = \Gamma_{(D,W)}(E). \tag{3}$$

Thus an extension E of (D, W) is a minimal set containing W which is closed under first-order consequence and for which none of the default rules can be applied to obtain a conclusion not already in E.

Reiter [102] shows that an extension can be defined in an alternative semi-inductive way.

Theorem 2.2 (Reiter [102]) *Let* (D, W) *be a default theory and* $E \subseteq \mathcal{L}$. *Define*

$$E_0 = W \tag{4}$$

and for all $i = 0, 1, \ldots,$

$$E_{i+1} = \mathrm{Cn}(E_i) \cup \mathrm{apply}_{D,E}(E_i). \tag{5}$$

Then E *is an extension of* (D, W) *if and only if*

$$E = \bigcup_{i=0}^{\infty} E_i. \tag{6}$$

Default logic is one of the most prominent formalizations of nonmonotonic reasoning. The books of Besnard [7], Etherington [25], and Marek and Truszczyński [76] are good sources on default logic. There is a body of results indicating that default logic can capture a large number of different forms of nonmonotonic reasoning. Default logic is especially closely related to reasoning in logic programs and deductive databases [10, 11, 32]. Default logic has been applied in formalizing inheritance reasoning and truth maintenance systems [23, 42]. Connections have been established between default logic and autoepistemic logic and McDermott and Doyle style nonmonotonic modal logics [51, 70, 71, 75, 118, 119], circumscription [24, 25], diagnosis [104], and abductive reasoning [94]. Default logic has not a uniform syntax but two categories of syntactical objects, first-order sentences and default rules. Hence representing defaults about defaults is not quite straightforward, for example. Also expressions saying that something is believed or disbelieved prove to be useful, in particular, for queries and integrity constraints [105] but are not available in default logic.

2.2 Nonmonotonic Modal Logics

McDermott and Doyle [82, 81] proposed a scheme for defining a nonmonotonic modal logic for a given monotonic modal logic S. The language $\mathcal{L}_{\mathrm{L}}$ of a classical modal logic S is an extension of that of the propositional calculus and includes a new monadic operator L. So e.g., $L(p \wedge (\neg Lq \rightarrow p))$ is a formula of $\mathcal{L}_{\mathrm{L}}$[1]. For example, Chellas [16]

[1] We use the following notations for the connectives of the propositional calculus: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (implication), $\leftrightarrow$ (equivalence).

and Hughes and Cresswell [40] are good sources on modal logics.

Given a classical monotonic modal logic S, a possible set of nonmonotonic conclusions from a set of premises is defined as a solution to a fixed point equation. The solutions, the fixed points, are called *S-expansions* of the premises Σ. We use the following notations: $L\Delta = \{L\phi \mid \phi \in \Delta\}$, $\neg\Delta = \{\neg\phi \mid \phi \in \Delta\}$, and $\overline{\Delta} = \mathcal{L}_{\mathrm{L}} - \Delta$. Hence $\neg L\overline{\Delta} = \{\neg L\phi \mid \phi \in \mathcal{L}_{\mathrm{L}} - \Delta\}$.

Definition 2.3 (McDermott and Doyle [82]) *Let S be a modal logic and $\Sigma \subseteq \mathcal{L}_{\mathrm{L}}$. A set of formulae Δ is an* S-expansion *of Σ if and only if*

$$\Delta = \{\phi \mid \Sigma \cup \neg L\overline{\Delta} \vdash_S \phi\} \tag{7}$$

where $\vdash_S$ is the derivability relation of the logic S.

The work of McDermott and Doyle [82, 81] was extented first by Shvarts [108] and then by Marek et al. [67]. A good source on nonmonotonic modal logics is the book of Marek and Truszczyński [76].

Statements of the type "birds typically fly" can be formalized in nonmonotonic modal logics using the formula

$$b \wedge \neg L\neg f \rightarrow f.$$

If the operator L is read "is believed", the formula states that if a bird is not believed not to fly, it flies.

Autoepistemic logic was introduced by Moore [84] as a reconstruction of McDermott and Doyle's [82] first attempt where the classical propositional calculus was used as the underlying logic. The aim was to avoid some deficiences of McDermott and Doyle's first logic. The difference is that Moore modified the fixed point equation (7) by adding a direct rule for positive introspection saying that if ϕ is belief, so is $L\phi$. The solutions of Moore's fixed point equation are called *stable expansions* of the premises.

Definition 2.4 (Moore [84]) *Let $\Sigma \subseteq \mathcal{L}_{\mathrm{L}}$. A set of formulae Δ is a* stable expansion *of Σ if and only if*

$$\Delta = \{\phi \mid \Sigma \cup L\Delta \cup \neg L\overline{\Delta} \vdash \phi\} \tag{8}$$

where $\vdash$ is the propositional consequence relation which treats $L\phi$ formulae like atomic formulae.

With direct positive introspection it is sufficient to consider only propositional consequences. In McDermott and Doyle style nonmonotonic logics positive introspection is achieved by using appropriate modal axioms and the necessitation rule.

The relationships between nonmonotonic modal logics and other forms of nonmonotonic reasoning have been investigated by several researchers. Gelfond [30] demonstrates how Moore's autoepistemic logic can be used to formalize commonsense

reasoning. Shvarts [108] shows that Moore's autoepistemic logic can be seen as a McDermott and Doyle style nonmonotonic modal logic. Marek et al. [66] extend the relationships between nonmonotonic modal logics and the connections to Moore's autoepistemic logic. The first results on the close relationship between autoepistemic logic and Reiter's default logic were obtained by Konolige [51] who proposes a translation between autoepistemic logic and default logic. This translation is studied further by Marek and Truszczyński [70, 75]. Several other embeddings of default logic into variants of autoepistemic logic, McDermott and Doyle style nonmonotonic modal logics and other modal systems have been studied [38, 47, 63, 64, 65, 72, 71, 88, 90, 112, 113, 119, 118]. Konolige [52] shows how to capture circumscription in autoepistemic logic. The connections between different semantics of logic programs and autoepistemic logic have been investigated intensively [12, 22, 29, 31, 46, 68, 69, 96, 99, 98]. Autoepistemic logic is shown to provide a logical basis for nonmonotonic truth maintenance systems [22, 27, 100]. Gelfond and Przymusinska [33] investigate the formalization of inheritance reasoning in autoepistemic logic. Kakas and Mancarella [45] suggest a connection between abductive reasoning and autoepistemic logic.

2.3 Circumscription

Circumscription introduced by McCarthy [78] is a powerful formalism which allows for the handling of handle incomplete and negative information. Circumscription gives a syntactical account of predicate minimization. Predicate circumscription conjectures for a set of predicates P that for p in P the entities for which we know that p holds are the only for which p holds. The closed world assumption [17] can be captured by predicate circumscription. Domain circumscription conjectures that the "known" entities are the only entities there are. It turns out that domain circumscription is subsumed by predicate circumscription.

Notation If P and Q are n-ary predicates, $P \leq Q$ denotes $\forall x(P(x) \rightarrow Q(x))$. This notation applies also to tuples of predicates, $P = P_1, P_2, ...P_n$ and $Q = Q_1, Q_2, ...Q_n$ assuming that P_i and Q_i have the same arity. Then $P \leq Q$ denotes $P_1 \leq Q_1 \wedge P_2 \leq Q_2 \wedge ... \wedge P_n \leq Q_n$. We write $P = Q$ for $P \leq Q \wedge Q \leq P$. $P < Q$ stands for $P \leq Q \wedge \neg(Q \leq P)$.

Definition 2.5 (Predicate Circumscription) *Given a tuple P of predicate constants $A(P)$, the* circumscription *of P in $A(P)$ is the formula $A(P) \wedge \neg\exists\Phi(A(\Phi) \wedge \Phi < P)$. We denote this formula by $Circ(A; P)$.*

Formula circumscription [79] makes it possible to assert that for predicate constants P and predicate or function constants Z occurring in a formula $A(P, Z)$, P has a minimal possible extension provided $A(P, Z)$ holds and Z is allowed to vary.

Definition 2.6 (Formula Circumscription) *Given a tuple P of predicate constants, a tuple Z of function resp. predicate constants disjoint from P and a formula $A(P, Z)$. The* circumscription *of P in $A(P, Z)$ with variables Z varying is the formula $A(P, Z) \wedge \neg\exists\Phi\exists\phi(A(\Phi, \phi) \wedge \Phi(x) < P)$. We denote this formula by $Circ(A; P; Z)$.*

This expresses that P has a minimal possible extension provided that $A(P;Z)$ holds and Z is allowed to vary.

Circumscription extends first-order logic. Circumscriptive conclusions from a formula A are the formulae that can be deduced classically from $Circ(A;P;Z)$.

Statements of the type "birds typically fly" can be formalized using circumscription in terms of exceptions. A bird flies unless it is exceptional:

$$\forall x((b(x) \wedge \neg exception(x)) \rightarrow f(x)).$$

The idea is to circumscribe *exception*, i.e. minimize exceptions to only those objects which are known to be exceptional.

Semantically, circumscription is characterized by minimal model semantics. A partial preorder between structures with respect to sets of predicates P and Z is defined as follows.

Given an interpretation I, let us denote by $\mid I \mid_P$ the extension of P in I.

Definition 2.7 *Given two interpretations I and J, $I <_{P,Z} J$ when $\mid I \mid_P \subset \mid J \mid_P$ and I and J have the same domain and the same extensions for the predicates not in P and in Z. We say also that I is smaller than J with respect to P.*
Given a set of interpretations S, an interpretation $I \in S$ is minimal in S with respect to $<_{P,Z}$ if there is no interpretation $I' \in S$ such that $I' <_{P,Z} I$.

The relation $<_{P,Z}$ is a preorder. It is reflexive and transitive, but not antisymmetric. It can be the case that $I <_{P,Z} J$ and $J <_{P,Z} I$ without $I = J$. This is due to the presence of variable predicates whose interpretations may differ in I and J.

McCarthy [78] shows the following correspondence theorem for minimal models:

Theorem 2.8 *Let A be a first-order theory and P, Z sets of predicate resp. predicate and function symbols. Then every $<_{P,Z}$-minimal model for A satisfies every instance of $Circ(A;P;Z)$.*

McCarthy [79] shows how to formalize many important forms of commonsense reasoning using circumscription. Circumscription is one of the most prominent approaches to reasoning about actions where the frame problem provides the key challenge [48, 61, 1, 2]. Circumscription has been used to capture, e.g., semantics of logic programs [60]. The relationships between default logic and circumscription as well as between autoepistemic logic and circumscription have been investigated by a number of researchers [24, 25, 41, 62, 52]. Etherington's [25] and Brewka's [13] books are good sources on circumscription.

3 Decidability and Complexity

In default logic possible sets of correct conclusions from a default theory are given in terms of extensions of the default theory and in nonmonotonic modal logics (and in

autoepistemic logic) in terms of expansions. It turns out that there are default theories with multiple extensions and similarly in nonmonotonic modal logics premises with multiple expansions. Hence the concepts of *cautious (skeptical)* and *brave (credulous)* reasoning are introduced. Given a class of extensions (expansions), a formula is derivable from a set of premises in the *cautious* sense if and only if it is in every extension (expansion) of the premises and in the *brave* sense if and only if it is at least in one of the extensions (expansions). So the fundamental decision problems for default logic and nonmonotonic modal logics are cautious and brave reasoning and the problem of deciding whether a set of premises has at least one extension/expansion (the *extension/expansion existence problem*). Circumscriptive conclusions from a set of premises can been seen as cautious conclusions because they are the formulae that are true in every minimal model of the premises.

For first-order default logic these problems are not even semi-decidable [102]. This holds also for first-order autoepistemic logic even when quantification into the scope of the modal operator is not allowed [87]. If quantification into the scope of the modal operator is allowed the problems are highly undecidable [53].

For propositional default logic extension existence problem and brave reasoning are Σ_2^p complete problems in the polynomial time hierarchy and cautious default reasoning is Π_2^p complete [37, 117]. Balcázar et al. [3] and Garey and Johnson [28] provide good sources of basic definitions and results on the polynomial time hierarchy and related issues in complexity theory. Kautz and Selman [49], Stillman [116], and Ben-Eliyahu and Dechter [4, 5] have obtained further complexity results on restricted subclasses of default theories.

For propositional nonmonotonic modal logics similar results hold: For autoepistemic logic based on Moore style expansions and L-hierarchic expansions [88] and for McDermott and Doyle style logics based on a number of modal logics expansion existence problem and brave reasoning are Σ_2^p complete problems and cautious reasoning is Π_2^p complete [37, 87, 89, 107]. Marek and Truszczyński [73, 74] present complexity results on restricted subclasses of autoepistemic reasoning.

The problem of deciding whether a propositional sentence is a circumscriptive consequence of a set of propositional sentences is also a Π_2^p complete problem [20]. Cadoli and Lenzerini [14] establish complexity results on limited subclasses of circumscriptive reasoning including some tractability results. General circumscription is highly uncomputable [106]. Several methods for automating circumscriptive reasoning in interesting subclasses have been proposed [59, 97, 36, 39]. Cadoli and Schaerf [15] have compiled a survey of the complexity results for nonmonotonic logics.

The completeness results with respect to the second level of the polynomial time hierarchy imply that there are two independent sources of complexity in nonmonotonic reasoning:

- required monotonic reasoning and
- conflict resolution.

The monotonic reasoning task refers to classical reasoning that is performed on the basis of the premises and the "default assumption" made during the reasoning

process and the conflict resolution task involves the problem of choosing a suitable set "default assumptions" from a set of available candidates which might be conflicting.

To obtain a subclass of nonmonotonic reasoning with favorable computational properties it is not enough to restrict the form of formulae to make the classical reasoning task easy as indicated by the results on restricted forms of nonmonotonic reasoning (see e.g. [49, 116, 73, 74]). The notion of stratification originally introduced in the context of logic programs offers an approach to limit the complexity of the conflict resolution problem. It has been shown that when considering only stratified sets of autoepistemic formulae [73] the conflict resolution task can be solved efficiently and a general iterative decision method can be devised [92]. The iterative decision method can be used also for stratified default logic, logic programs, truth maintenance systems, and McDermott and Doyle style nonmonotonic modal logics. For example, a linear time decision procedure for stratified propositional logic programs and truth maintenance systems is obtained [92].

4 Relationships

Relationships between default logic and nonmonotonic modal logics have been studied widely and they have turned out to be closely related. One of the first results was Konolige's study on the relations between autoepistemic logic and default logic [51] which was extended by Marek and Truszczyński [70]. Also the relationship between McDermott and Doyle style nonmonotonic modal logics and default logics have been studied intensively [71, 118, 119]. These results show that default logic can be embedded into nonmonotonic modal logics using local translations in the sense that extensions of a default theory are expansions of the translated theory. Imielinski [41] points out that default rules cannot not be translated modularly into circumscription. This due to the fact that circumscription cannot capture the notion of provability or logical consequence required to translate a default rule.

As an example we consider a particular intuitively appealing embedding (9) of default logic into nonmonotonic modal logics introduced by Truszczyński [118].

$$\mathrm{AE}_{\mathrm{DL}}(\frac{a : b_1, \ldots, b_n}{c}) = \\ (La \wedge L\neg L\neg b_1 \wedge \cdots \wedge L\neg L\neg b_n) \rightarrow c \tag{9}$$

$$\mathrm{AE}_{\mathrm{DL}}(D, W) = W \cup \{\mathrm{AE}_{\mathrm{DL}}(d) \mid d \in D\} \tag{10}$$

Truszczyński [118] shows that the consistent expansions of the translated default theory in any McDermott and Doyle style nonmonotonic modal logics between **N** and **S4F** correspond to the extensions of the default theory.

Theorem 4.1 (Truszczyński [118]) *Let (D, W) be a default theory where D is finite. Then a consistent set $E \subseteq \mathcal{L}$ is an extension of (D, W) if and only if $E = \Delta \cap \mathcal{L}$ for an S-expansion Δ of $\mathrm{AE}_{\mathrm{DL}}(D, W)$ for any modal logic S between* **N** *and* **S4F**.

Default logic is closely related to the answer set semantics [32] of extended logic programs, i.e. logic programs where classical negation ($\neg$) is allowed even in the scope of the negation-as-failure operator (*not*). Gelfond and Lifschitz [32] show that answer sets can be characterized using default logic by translating rules in an extended logic program in the following way.

$$\mathrm{LP}_{\mathrm{DL}}(l_0 \leftarrow l_1, \ldots, l_m, not\, l_{m+1}, \ldots, not\, l_n) = \frac{l_1 \wedge \cdots \wedge l_m : \overline{l_{m+1}}, \ldots, \overline{l_n}}{l_0} \tag{11}$$

where $\overline{l}$ is the literal complementary to l: $\overline{a} = \neg a$ and $\overline{\neg a} = a$. For a set of rules P, $\mathrm{LP}_{\mathrm{DL}}(P) = \{\mathrm{LP}_{\mathrm{DL}}(r) \mid r \in P\}$.

Theorem 4.2 (Gelfond and Lifschitz [32]) *Let P be an extended logic program. Then S is an answer set of P if and only if S is the set of literals in an extension E of $\mathrm{LP}_{\mathrm{DL}}(P)$.*

For logic programs where classical negation is not allowed, i.e. general logic programs, the answer set semantics coincides with the stable model semantics [31]. Elkan [22] has shown that given a set of justifications J the grounded model of J computed by a truth maintenance system is just a stable model of J when J is seen as a propositional logic program. Hence default logic can capture truth maintenance systems.

In default logic, autoepistemic logic and other nonmonotonic modal logics it is possible to capture minimal models of a set of premises by adding new expressions to that effect. For simplicity we consider the propositional case and a notion of minimal models where the models are minimized with respect to a given set P of atomic sentences keeping another set F of atomic sentences fixed and letting all the other atomic sentences vary during the minimization. Propositional interpretations can be seen as sets of atomic sentences and the corresponding order relation on the propositional interpretations can be defined as follows given the sets P and F of atoms. A model I of Σ is $\langle P, F\rangle$-*minimal* if there is no model I' of Σ containing the same fixed atomic sentences from F but for which the atomic sentences in P are properly included in the atomic sentences from P in I. This notion of minimality is a propositional version of the minimality in parallel circumscription [59] where the atomic sentences P are minimized and the atomic sentences not in F or P are allowed to vary. Gelfond and et al. [34] show that parallel circumscription corresponds to a particular form of the negation as failure rule (the extended closed world assumption).

For a set of formulae P, we define

$$\mathrm{MIN}_{\mathrm{DL}}(P) = \{\frac{: \overline{\phi}}{\overline{\phi}} \mid \phi \in P\}, \tag{12}$$

$$\mathrm{FIX}_{\mathrm{DL}}(P) = \mathrm{MIN}_{\mathrm{DL}}(P) \cup \mathrm{MIN}_{\mathrm{DL}}(\neg P). \tag{13}$$

Theorem 4.3 *Let P and F be sets of atomic sentences and $\Sigma \subseteq \mathcal{L}$. For every sentence $\phi \in \mathcal{L}$, ϕ is true in every $\langle P, F\rangle$-minimal model of Σ if and only if ϕ is in every extension of $(\mathrm{MIN}_{\mathrm{DL}}(P) \cup \mathrm{FIX}_{\mathrm{DL}}(F), \Sigma)$.*

A similar result holds also, e.g., for autoepistemic logic based on Moore style expansions. For a set of formulae P, we define

$$\mathrm{MIN}_{\mathrm{AE}}(P) = \{\neg L\phi \rightarrow \overline{\phi} \mid \phi \in P\}, \quad (14)$$

$$\mathrm{FIX}_{\mathrm{AE}}(P) = \mathrm{MIN}_{\mathrm{AE}}(P) \cup \mathrm{MIN}_{\mathrm{AE}}(\neg P). \quad (15)$$

Theorem 4.4 *Let P and F be finite sets of atomic sentences and $\Sigma \subseteq \mathcal{L}$. For every $\phi \in \mathcal{L}$, ϕ is true in every $\langle P, F\rangle$-minimal model of Σ if and only if ϕ is in every stable expansion of $\Sigma \cup \mathrm{MIN}_{\mathrm{AE}}(P) \cup \mathrm{FIX}_{\mathrm{AE}}(F)$.*

Etherington [23] reports similar results of minimization using default logic in the first-order case but without any fixed predicates and with a finite, fixed domain. Konolige [52] shows how to achieve minimization in the case of non-finite domains using a variant of autoepistemic logic which allows quantification into the scope of L operators. He employs the schemes (14) and (15) to minimize and fix predicates.

5 Proof Systems for Nonmonotonic Logics

5.1 Nonmonotonic Modal Logics

Expansions are infinite sets of sentences and a finitary characterization is needed to enable a computational treatment of reasoning in nonmonotonic modal logics. A key observation is that expansions are uniquely determined by $L\phi$ subformulae of the premises. For autoepistemic logic a characterization based on this principle has been developed by Niemelä [87] and for several McDermott and Doyle style nonmonotonic modal logic by Schwarz, Marek and Truszczyński [108, 67].

If each expansion has a finite characterizing set, a straightforward decision method can be devised. For brave reasoning an expansion containing a given formula must be found and for cautious reasoning it must be verified that a given formula is in all the expansions. This can be achieved by iterating over all possible candidates for characterizing sets: if the candidate satisfies the conditions for a characterizing set, then it is tested whether the given formula is contained in the expansion induced by the characterizing set. Figure 1 presents an outline of such a decision method. The method returns a positive answer for brave derivability if an expansion is found containing the given formula and for cautious derivability when the formula is in every expansion. This kind of a method requires that tests for

- a set to qualify as a characterizing set and
- membership in an expansion, i.e. deciding whether a formula belongs to an expansion when given only the characterizing set of the expansion,

must be computable. Using this kind of an approach decidability of autoepistemic logic based on several types of expansions and a number of nonmonotonic modal

INPUT: A finite set of sentences Σ and a sentence ϕ.

C-answer:= TRUE; B-answer:= FALSE;
FOR every subset S of $L\psi$ subformulae of Σ **DO**
  IF S yields a characterizing set K for an expansion **THEN**
    IF ϕ belongs to the expansion induced by K **THEN**
      B-answer:= TRUE
    ELSE
      C-answer:= FALSE.

Figure 1: A Straightforward Decision Method for Nonmonotonic Modal Logics

logics have been proved [87, 89, 67]. However, this kind of a straightforward decision method is not very efficient because there might a huge number of potential candidates for characterizing sets. The method iterates over every subset of the set of $L\chi$ subformulae of the premises. If the cardinality of a set is n, it has 2^n subsets. Moreover, each iteration might be computationally costly because of the tests for characterizing sets and membership in expansions.

The structure of the space of nonmonotonic modal logics has turned out to be quite regular. Marek et al. [67] offer a unified treatment of McDermott and Doyle style nonmonotonic modal logics in the propositional case. They show that the underlying monotonic modal logic plays a secondary role in the definition of nonmonotonic reasoning based on the fixed point equation (7): several monotonic modal logics induce the same nonmonotonic system, i.e. expansions coincide for a range of underlying modal logics. For example, for every modal logic S between **5** and **KD45**, an S-expansion is a stable expansion. If the set of premises is taken from a restricted class of formulae, even larger ranges exist. For example, let Σ be strongly stratified [67]. Then for every modal logic S such that $\mathbf{N} \subseteq S$ and $S \subseteq \mathbf{KD45}$ or $S \subseteq \mathbf{SW5}$, Δ is an S-expansion if and only if Δ is the unique stable expansion of Σ [67].

All nonmonotonic modal logics based on the fixed point equation (7) share a property that so called weakly grounded (or circular grounded) beliefs are not excluded from the expansions. For example, $\{\neg L\neg Lp \rightarrow p\}$ has an **N**-expansion[2] containing a circular grounded belief p. This property of **N**-expansions is immediately inherited by other McDermott and Doyle style nonmonotonic modal logics because of the following property of nonmonotonic modal logics: for any modal logics S_1, S_2 contained in **S5**, if S_1 is contained in S_2, then any S_1-expansion is an S_2-expansion [67].

5.1.1 A Case Study: Autoepistemic Reasoning

In this section we consider as a case study autoepistemic logic and show how to develop more efficient decision methods. We concentrate on autoepistemic logic based on *L-hierarchic expansions* [88] for several reasons. Firstly, in L-hierarchic

[2]The modal logic **N** is an extension of the propositional calculus with no modal axioms but the rule of necessitation.

expansions the circularly grounded beliefs are eliminated. Secondly, no modal axioms are needed but only classical logic and the direct rules for positive and negative introspection. Thirdly, autoepistemic logic based on L-hierarchic expansions offers a unified framework for nonmonotonic reasoning covering default logic, circumscription, and logic programs with classical negation using simple local mappings.

We consider a general setting where a classical calculus CL is given and an autoepistemic logic $\mathrm{CL}_{\mathrm{ae}}$ is built on top of that. This kind of a general setting has already been studied [89]. Let $\mathcal{L}$ be the language of CL. We extend $\mathcal{L}$ by adding a monadic operator L not appearing in $\mathcal{L}$ and obtain an autoepistemic language $\mathcal{L}_{\mathrm{L}}$ which is the language of $\mathrm{CL}_{\mathrm{ae}}$. The language $\mathcal{L}_{\mathrm{L}}$ is defined recursively as $\mathcal{L}$ but with an extra formation rule: if $\phi \in \mathcal{L}_{\mathrm{L}}$ is a sentence, then $L\phi \in \mathcal{L}_{\mathrm{L}}$. Thus we consider autoepistemic logics where quantification into a modal context is not allowed even if $\mathcal{L}$ contains quantifiers (e.g. formulae of the form $\forall x LP(x)$ are not allowed). To simplify the treatment we consider only closed formulae. So from here on we let $\mathcal{L}$ and $\mathcal{L}_{\mathrm{L}}$ contain only closed formulae. In the general setting an ideally rational agent believes the logical consequences given by a consequence relation $\models_{\mathrm{ae}}$ which is a simple extension of the consequence relation $\models$ of CL where the $L\phi$ formulae are treated like atomic formulae in the propositional calculus.

The idea in L-hierarchic expansions is to build a set of beliefs from premises Σ by applying introspection to sentences in the order given by an enumeration ε. A set $\mathbf{B}^{\varepsilon}(\Sigma)$ is constructed which contains all the results of introspection. The set $\mathbf{B}^{\varepsilon}(\Sigma)$ together with Σ induces the set of beliefs $\mathbf{SE}^{\varepsilon}(\Sigma)$ of an agent having initial assumptions Σ after introspecting sentences in the order ε.

Definition 5.1 (Niemelä [89]) *Let $\Sigma \subseteq \mathcal{L}_{\mathrm{L}}$. Let $\varepsilon = \psi_1, \psi_2, \ldots$ be an enumeration of sentences in $\mathcal{L}_{\mathrm{L}}$. Let $\mathbf{B}^{\varepsilon}_0(\Sigma) = \emptyset$ and define $\mathbf{B}^{\varepsilon}_{i+1}(\Sigma)$ for $i = 0, 1, \ldots$ as follows:*

$$\mathbf{B}^{\varepsilon}_{i+1}(\Sigma) = \begin{cases} \mathbf{B}^{\varepsilon}_i(\Sigma) \cup \{L\psi_{i+1}\} \text{ if } \Sigma \cup \mathbf{B}^{\varepsilon}_i(\Sigma) \models_{\mathrm{ae}} \psi_{i+1} \\ \mathbf{B}^{\varepsilon}_i(\Sigma) \cup \{\neg L\psi_{i+1}\} \text{ otherwise} \end{cases}$$

Finally let

$$\begin{aligned} \mathbf{B}^{\varepsilon}(\Sigma) &= \bigcup_{i=0}^{\infty} \mathbf{B}^{\varepsilon}_i(\Sigma) \\ \mathbf{SE}^{\varepsilon}(\Sigma) &= \{\phi \in \mathcal{L}_{\mathrm{L}} \mid \Sigma \cup \mathbf{B}^{\varepsilon}(\Sigma) \models_{\mathrm{ae}} \phi\}. \end{aligned}$$

To guarantee that the introspection principles of autoepistemic reasoning are respected the enumeration ε is required to be *acceptable* with respect to the set of premises. To ensure that also formulae with nested belief operators are tightly grounded in the premises enumerations are required to be *L-hierarchic*.

Definition 5.2 *An enumeration ε is* Σ-acceptable *if there is no i and no formula ϕ such that $\neg L\phi \in \mathbf{B}^{\varepsilon}_i(\Sigma)$ but $\Sigma \cup \mathbf{B}^{\varepsilon}_i(\Sigma) \models_{\mathrm{ae}} \phi$.*

Definition 5.3 *Let ε be an enumeration $\psi_1, \psi_2, \ldots$ of sentences in $\mathcal{L}_{\mathrm{L}}$. The enumeration ε is* L-hierarchic *if and only if for all ψ_i, ψ_j in ε holds that if $L\psi_i$ is a subformula of ψ_j, then $i < j$.*

A set of sentences Δ is an L-hierarchic expansion *of Σ if and only if $\Delta = \mathbf{SE}^{\varepsilon}(\Sigma)$ for some Σ-acceptable L-hierarchic enumeration ε of $\mathcal{L}_{\mathrm{L}}$.*

L-hierarchic expansions provide a tightly grounded subclass of Moore style expansions where no dependence on the syntactic representation of the premises is introduced.

Autoepistemic logic based on L-hierarchic expansions contains as special cases other prominent forms of nonmonotonic reasoning. Default logic is captured using the translation (9).

Theorem 5.4 (Niemelä [90]) *Let (D, W) be a default theory where D is finite. Then a set $E \subseteq \mathcal{L}$ is an extension of (D, W) if and only if $E = \Delta \cap \mathcal{L}$ for an L-hierarchic expansion Δ of $\mathrm{AE}_{\mathrm{DL}}(D, W)$.*

Default logic is closely related to the answer set semantics [32] of extended logic programs. Consequently, the answer set semantics can be embedded into autoepistemic logic using the following mapping.

$$\begin{aligned}&\mathrm{LP}_{\mathrm{AE}}(l_0 \leftarrow l_1, \ldots, l_m, \mathit{not}\, l_{m+1}, \ldots, \mathit{not}\, l_n) = \\ &\quad Ll_1 \wedge \ldots \wedge Ll_m \wedge L\neg Ll_{m+1} \wedge \ldots \wedge L\neg Ll_n \rightarrow l_0 \end{aligned} \tag{16}$$

Theorem 5.5 (Niemelä [90]) *Let P be a finite extended logic program. Then S is an answer set of P if and only if S is the set of literals in an L-hierarchic expansion of $\mathrm{LP}_{\mathrm{AE}}(P)$.*

For general logic programs the answer set semantics coincides with the stable model semantics [31].

Corollary 5.6 *Let P be a finite general logic program. Then S is a stable model of P if and only if S is the set of atomic sentences in an L-hierarchic expansion of $\mathrm{LP}_{\mathrm{AE}}(P)$.*

Elkan's [22] result on the correspondence between grounded models computed by a truth maintenance system and stable models implies that truth maintenance systems can be seen as a special case of autoepistemic logic based on L-hierarchic expansions.

Corollary 5.7 *Let J be a finite set of justifications. Then M is a grounded model of J if and only if M is the set of atomic sentences in an L-hierarchic expansion of $\mathrm{LP}_{\mathrm{AE}}(J)$.*

Also circumscription can be captured L-hierarchic expansions and Theorem 4.4 holds also for L-hierarchic expansions.

5.1.2 A Finitary Characterization of Autoepistemic Expansions

We start our presentation of the computational treatment of autoepistemic reasoning by explaining how each autoepistemic expansion, i.e. a Moore style and L-hierarchic expansion, can be characterized using a finite set. We use a characterization based on *full sets* [87, 89]. The notion of full sets provides a simple and compact representation of expansions which is straightforwardly applicable in the general setting considered here. An overview of other approaches to characterizing stable expansions can be found in [89].

First we introduce some notation. For an $L\chi$ formula, q-atom$(L\chi)$ = q-atom$(\neg L\chi)$ = $L\chi$ and for a set Λ, q-atom(Λ) = $\{$q-atom$(\phi) \mid \phi \in \Lambda\}$. The set of all subformulae of the form $L\chi$ of a formula ϕ is denoted by $\mathrm{Sf}^{\mathrm{L}}(\phi)$ and $\mathrm{Sf}^{\mathrm{qL}}(\phi)$ is the set of all subformulae of the form $L\chi$ of ϕ which are not in the scope of another L operator in ϕ. We define a simple consequence relation $\models_{\mathrm{L}}$ which is given recursively on top of the underlying consequence relation $\models_{\mathrm{ae}}$. It turns out that this consequence relation corresponds exactly to membership in an expansion when given its characterizing full set.

Definition 5.8 *Given a set of sentences Σ and a sentence ϕ,*

$$\Sigma \models_{\mathrm{L}} \phi \text{ if and only if } \Sigma \cup \mathrm{SB}_{\Sigma}(\phi) \models_{\mathrm{ae}} \phi$$

where $\mathrm{SB}_{\Sigma}(\phi) = \{L\chi \in \mathrm{Sf}^{\mathrm{qL}}(\phi) \mid \Sigma \models_{\mathrm{L}} \chi\} \cup \{\neg L\chi \in \neg\mathrm{Sf}^{\mathrm{qL}}(\phi) \mid \Sigma \not\models_{\mathrm{L}} \chi\}$.

A finitary characterization for Moore style expansions can be provided in terms of sets of sentences constructed from the $L\chi$ and $\neg L\chi$ subformulae of the premises satisfying a special fullness condition. A full set serves as the kernel of a stable expansion; it uniquely characterizes the stable expansion. In fact there is a one-to-one correspondence between full sets and expansions.

Definition 5.9 *For a set of sentences Σ, a set $\Lambda \subseteq \mathrm{Sf}^{\mathrm{L}}(\Sigma) \cup \neg\mathrm{Sf}^{\mathrm{L}}(\Sigma)$ is Σ-full if and only if the following two conditions hold for every $L\chi \in \mathrm{Sf}^{\mathrm{L}}(\Sigma)$:*

1. $\Sigma \cup \Lambda \models_{\mathrm{ae}} \chi$ *if and only if* $L\chi \in \Lambda$.
2. $\Sigma \cup \Lambda \not\models_{\mathrm{ae}} \chi$ *if and only if* $\neg L\chi \in \Lambda$.

Theorem 5.10 (Niemelä [89]) *Let Σ be a set of sentences of $\mathcal{L}_{\mathrm{L}}$. Then a function* SE_{Σ} *defined as*

$$\mathrm{SE}_{\Sigma}(\Lambda) = \{\phi \in \mathcal{L}_{\mathrm{L}} \mid \Sigma \cup \Lambda \models_{\mathrm{L}} \phi\}$$

gives a bijective mapping from the set of Σ-full sets to the set of Moore style stable expansions of Σ and for a Σ-full set Λ, $\mathrm{SE}_{\Sigma}(\Lambda)$ is the unique stable expansion Δ of Σ such that $\Lambda \subseteq L\Delta \cup \neg L\overline{\Delta}$.

Example 2 The premise $\Sigma = \{\neg L\neg Lp \rightarrow p\}$ has two Σ-full sets $\Lambda_1 = \{L\neg Lp, \neg Lp\}$ and $\Lambda_2 = \{\neg L\neg Lp, Lp\}$. For example, Λ_1 is Σ-full as $\Sigma \cup \Lambda_1 \models_{\mathrm{ae}} \neg Lp$ and $\Sigma \cup \Lambda_1 \not\models_{\mathrm{ae}} p$. This implies that Σ has exactly two Moore style expansions $\mathrm{SE}_{\Sigma}(\Lambda_1)$ and $\mathrm{SE}_{\Sigma}(\Lambda_2)$. As $\Sigma \cup \Lambda_1 \not\models_{\mathrm{L}} p$, $p \notin \mathrm{SE}_{\Sigma}(\Lambda_1)$. However, $\neg LLp \in \mathrm{SE}_{\Sigma}(\Lambda_1)$ as $\Sigma \cup \Lambda_1 \models_{\mathrm{L}} \neg LLp$. ■

A finitary characterization for enumeration-based expansions is obtained by observing that full sets corresponding to enumeration-based (L-hierarchic) expansions are exactly those generated by acceptable (L-hierarchic) enumerations. First we note that for a finite enumeration the test for acceptability can be stated in the following simple form.

Proposition 5.11 (Niemelä [89]) *Let $\varepsilon_n = \psi_1, \ldots, \psi_n$ be an enumeration of a finite set. Then ε_n is Σ-acceptable if and only if ε_n satisfies the condition: for all $\neg L\phi \in \mathbf{B}_n^{\varepsilon_n}(\Sigma), \Sigma \cup \mathbf{B}_n^{\varepsilon_n}(\Sigma) \not\models_{\mathrm{ae}} \phi$.*

Theorem 5.12 (Niemelä [89]) *Let $\Sigma \subseteq \mathcal{L}_{\mathrm{L}}$ be a set of sentences such that $\mathrm{Sf}^{\mathrm{L}}(\Sigma)$ is finite. If an enumeration $\varepsilon_n = \psi_1, \ldots, \psi_n$ of $\{\phi \mid L\phi \in \mathrm{Sf}^{\mathrm{L}}(\Sigma)\}$ is Σ-acceptable, then $\mathbf{B}_n^{\varepsilon_n}(\Sigma)$ is a Σ-full set and there exists a Σ-acceptable enumeration ε of $\mathcal{L}_{\mathrm{L}}$ such that $\mathbf{SE}^{\varepsilon}(\Sigma)$ is the unique stable expansion Δ of Σ for which $\mathbf{B}_n^{\varepsilon_n}(\Sigma) \subseteq L\Delta \cup \neg L\overline{\Delta}$ (i.e. $\mathbf{SE}^{\varepsilon}(\Sigma) = \mathrm{SE}_{\Sigma}(\mathbf{B}_n^{\varepsilon_n}(\Sigma))$). Furthermore, if ε_n is L-hierarchic, then ε is also.*

Theorem 5.13 (Niemelä [89]) *Let $\Sigma \subseteq \mathcal{L}_{\mathrm{L}}$ be a set of sentences. For every Σ-acceptable (L-hierarchic) enumeration ε of $\mathcal{L}_{\mathrm{L}}$, there exists a Σ-acceptable (L-hierarchic) enumeration ε_L of $\{\phi \mid L\phi \in \mathrm{Sf}^{\mathrm{L}}(\Sigma)\}$ such that $\mathbf{B}^{\varepsilon_L}(\Sigma)$ is the unique Σ-full set Λ for which $\Lambda \subseteq L\mathbf{SE}^{\varepsilon}(\Sigma) \cup \neg L\overline{\mathbf{SE}^{\varepsilon}(\Sigma)}$ holds and thus $\mathbf{SE}^{\varepsilon}(\Sigma) = \mathrm{SE}_{\Sigma}(\mathbf{B}^{\varepsilon_L}(\Sigma))$.*

Example 3 The Moore style expansions in Example 2 are both characterizable using enumerations of $\{\phi \mid L\phi \in \mathrm{Sf}^{\mathrm{L}}(\Sigma)\} = \{p, \neg Lp\}$: $\mathrm{SE}_{\Sigma}(\Lambda_1)$ by the enumeration $\varepsilon_2 = p, \neg Lp$ and $\mathrm{SE}_{\Sigma}(\Lambda_2)$ by the enumeration $\varepsilon_2' = \neg Lp, p$, i.e. $\Lambda_1 = \mathbf{B}_2^{\varepsilon_2}(\Sigma)$ and $\Lambda_2 = \mathbf{B}_2^{\varepsilon_2'}(\Sigma)$. Only ε_2 is L-hierarchic. Hence Σ has exactly two enumeration-based expansions but only one of them is L-hierarchic, i.e. $\mathrm{SE}_{\Sigma}(\mathbf{B}_2^{\varepsilon_2}(\Sigma))$. ■

5.1.3 Approaches to Automating Autoepistemic Reasoning

By the aforementioned results the straightforward decision method in Figure 1 provides a decision method for autoepistemic reasoning in the general setting provided that the consequence relation $\models_{\mathrm{ae}}$ given by the underlying classical logic is decidable. Clearly the conditions for characterizing sets are computable if the underlying consequence relation $\models_{\mathrm{ae}}$ is decidable and by Theorem 5.10 the membership tests are computable using the consequence relation $\models_{\mathrm{L}}$ which reduces to consequence tests for $\models_{\mathrm{ae}}$ (see [89] for details).

Decision problems in propositional autoepistemic reasoning are complete problems with respect to the second level of the polynomial time hierarchy [89] which implies that any general decision method for autoepistemic reasoning most likely has an exponential worst-case *time* complexity. However, a practically oriented decision method should be able to take advantage of possible regularities in a given set of premises. For example, consider a set of premises

$$\Sigma = \{p_0, Lp_0 \rightarrow p_1, Lp_1 \rightarrow p_2, \ldots, Lp_{n-1} \rightarrow p_n\}. \tag{17}$$

The straightforward approach would examine the 2^n subsets of $\{Lp_0, \ldots, Lp_{n-1}\}$ to find the unique Σ-full set $\Lambda = \{Lp_0, \ldots, Lp_{n-1}\}$. However, by carefully exploiting the monotonicity of the underlying consequence relation $\models_{\mathrm{ae}}$ it is possible to find the full set without iterating over the subsets: for any Σ-full set Λ, $\Sigma \cup \Lambda \models_{\mathrm{ae}} p_0$ holds and thus $Lp_0 \in \Lambda$ must hold which in turn implies $Lp_1 \in \Lambda$ and so on.

Several decision methods have been proposed for autoepistemic reasoning. Some of these [85, 115, 66] are straightforward applications of a particular finitary characterization of autoepistemic reasoning, some [86] use extensively a specific underlying proof method and some [73, 44, 74] are based on the idea of mapping a decision problem in autoepistemic logic into another problem.

The advanced methods include techniques where nonmonotonic reasoning is reduced to another problem such as a truth maintenance problem [44], a theorem proving problem [74], or a constraint satisfaction problem [4]. These reductions provide valuable insights to the relationships between the corresponding problems and enable new techniques to be used for solving nonmonotonic reasoning problems. However, while there are interesting subclasses of nonmonotonic reasoning problems where the problem size increases only polynomially as a result of the reduction mapping (see, e.g. [74, 4]), in the general case the reductions can lead to an exponential increase in the problem size. Hence the worst case *space* complexity becomes exponential. Moreover, the reductions are computationally complex.

As an example we consider the approach of Junker and Konolige [44]. In this approach the premises are required to be in a normal form with no nested L operators. In the worst case, if a set of premises is transformed into the normal form, its size might increase exponentially. Then the premises are mapped to a set of justifications in a truth maintenance system. The size of the resulting set of justifications can be exponential with respect to the size of the premises and the mapping is computationally rather complex. This is because the set of justifications corresponding to premises Σ includes all monotonic justifications $\phi \leftarrow \omega_1, \ldots, \omega_n$ where $L\phi \in \mathrm{Sf}^{\mathrm{L}}(\Sigma)$ and $\{\omega_1, \ldots, \omega_n\}$ is a minimal *proof* of ϕ from $H = \{\omega \mid \neg L\alpha \vee L\beta_1 \vee \ldots \vee L\beta_m \vee \omega \in \Sigma - \mathcal{L}\}$, i.e. a minimal subset of H such that $(\Sigma \cap \mathcal{L}) \cup \{\omega_1, \ldots, \omega_n\} \models \phi$ holds. For an $L\phi$ subformula of the premises, ϕ can possess a very large number of minimal proofs. The problem of finding a proof of ϕ from a hypothesis set H given a background theory Σ, i.e. finding a set of $H' \subseteq H$ such that $\Sigma \cup H' \models \phi$, is closely related to logic-based abduction [21]. If the background theory is a set of clauses, then deciding whether there is a consistent proof of an atom from a set of atoms is Σ_2^p-complete in the propositional case [21]. As propositional brave autoepistemic reasoning is also Σ_2^p-complete [89], the search problem of finding a consistent proof of an atom is at least as hard as brave autoepistemic reasoning.

In Junker and Konolige's approach an expansion corresponds to a suitable model of the justifications. For example, strongly grounded expansions correspond to grounded models but deciding whether there is a grounded model is **NP**-complete [22]. The exponential worst case space complexity and the high time complexity of the reductions raise questions about the usability of the reduction approaches as general decision methods for nonmonotonic reasoning.

5.1.4 A Decision Method for Autoepistemic Reasoning

Here we present a recently developed decision method for autoepistemic reasoning [91]. The method works in a general setting, i.e. for an autoepistemic logic based on a given classical logic. The method does not commit to a particular underlying proof method and it provides a conceptually simple approach where the sources of complexity and targets for optimization are clearly identifiable. Furthermore, the method avoids exponential worst-case space requirements which seem to be inherent in other advanced methods [86, 73, 44, 74].

The decision method is presented for autoepistemic reasoning based on L-hierarchic expansions but it can be easily modified to cover also other classes of expansions (see [91] for details). The decision method builds Σ-full sets for L-hierarchic expansions of Σ. The Σ-full sets can be found by examining the enumerations of the set $\{\phi \mid L\phi \in \mathrm{Sf}^{\mathrm{L}}(\Sigma)\}$. However, the enumerations are not important but the sets of beliefs induced by the enumerations. This observation cuts the combinatorial explosion considerably as typically there is a large number of enumerations inducing the same Σ-full set. The full sets are developed together as far as possible so that the same computation is not repeated by exploiting the monotonicity of the underlying consequence relation $\models_{\mathrm{ae}}$. The novel decision method for autoepistemic reasoning needs only a theorem prover for the underlying monotonic consequence relation $\models_{\mathrm{ae}}$ as a subroutine.

Figure 2 presents the function $\mathbf{der}_{\mathrm{LE}}$ which is the key part of the novel decision method. The function $\mathbf{der}_{\mathrm{LE}}$ is a skeleton for the decision procedures for brave and cautious reasoning as well as for checking the existence of expansions. For that purpose it contains an unspecified function **test**. By changing this function the various decision procedures are obtained. The function $\mathbf{der}_{\mathrm{LE}}$ takes as input a set of premises Σ, sets B and F which give the common part of the Σ-full sets to be considered and a sentence ϕ which is just passed as an argument to the function **test**.

The purpose of $\mathbf{der}_{\mathrm{LE}}$ is to return true if and only if there exists an L-hierarchic expansion Δ of Σ containing $B \cup F$ such that $\mathbf{test}(\Sigma, \Lambda, \phi)$ returns true where Λ is the Σ-full set corresponding Δ, i.e. $\Delta = \mathrm{SE}_{\Sigma}(\Lambda)$. This is achieved by constructing Σ-full sets which contain $B \cup F$ and correspond to L-hierarchic expansions until a full set Λ is found for which $\mathbf{test}(\Sigma, \Lambda, \phi)$ returns true.

When explaining the method we use the following two notions. An enumeration ε is *fully L-hierarchic* if for every formula ϕ in ε for each subformula $L\chi$ of ϕ, χ appears before ϕ in ε. The set B is *generated* by some enumeration ε if $B = \mathbf{B}^{\varepsilon}(\Sigma)$.

In the decision method the idea is to expand the set B so that finally it contains every formula in a full set. By Theorem 5.13 for every L-hierarchic expansion, the corresponding full set is generated by some fully L-hierarchic enumeration from Σ. This implies that when deciding how to extend B the number of possibilities can be reduced by considering only those extensions which are generated by some fully L-hierarchic enumeration. Hence in the decision method B is extended only if the resulting set B' is generated by some fully L-hierarchic enumeration ε_m of a subset of $\{\phi \mid L\phi \in \mathrm{Sf}^{\mathrm{L}}(\Sigma)\}$ from Σ, i.e. $B' = \mathbf{B}^{\varepsilon_m}_m(\Sigma)$. The set F contains

FUNCTION $\mathbf{der}_{\mathrm{LE}}(\Sigma, B, F, \phi)$
INPUT: Finite sets of sentences Σ, B, and F and
a sentence ϕ.

$\langle B, F\rangle := \mathbf{pos}_{\mathrm{LE}}(\Sigma, B, F)$;
IF for some $\neg L\chi \in B$, $\Sigma \cup B \cup F \models_{\mathrm{ae}} \chi$ **THEN**
return false
ELSE IF $\mathrm{Sf}^{\mathrm{L}}(\Sigma) \subseteq$ q-atom$(B) \cup F$ **THEN**
IF $F \subseteq B$ **THEN**
return $\mathbf{test}(\Sigma, B, \phi)$
ELSE
return false
ELSE IF for some $L\chi \in \mathrm{Sf}^{\mathrm{L}}(\Sigma) -$ (q-atom$(B) \cup F$)
$\mathrm{Sf}^{\mathrm{L}}(\chi) \subseteq$ q-atom(B) **THEN**
IF $\mathbf{der}_{\mathrm{LE}}(\Sigma, B \cup \{\neg L\chi\}, F, \phi)$ returns false **THEN**
return $\mathbf{der}_{\mathrm{LE}}(\Sigma, B, F \cup \{L\chi\}, \phi)$
ELSE
return true
ELSE
return false.

Figure 2: A Skeleton for the Decision Procedures for Autoepistemic Reasoning Based on L-hierarchic Expansions

$L\chi$ sentences which should be included in the full set to be constructed but are "frozen": a sentence $L\chi \in F$ is added to B only after B *covers* all $L\chi'$ subformulae of χ, i.e. $\mathrm{Sf}^{\mathrm{L}}(\chi) \subseteq$ q-atom(B), and $\Sigma \cup B \models_{\mathrm{ae}} \chi$ holds. This ensures that B remains generatable from Σ by a fully L-hierarchic enumeration.

In $\mathbf{der}_{\mathrm{LE}}$ the results of positive introspection are first propagated as far as possible using the function $\mathbf{pos}_{\mathrm{LE}}$ presented in Figure 3. The function exploits the monotonicity of the underlying consequence relation $\models_{\mathrm{ae}}$ and expands the sets B and F, i.e. if B is included in a full set and $\Sigma \cup B \models_{\mathrm{ae}} \phi$, then $L\phi$ belongs to the full set.

Example 4 (i) Consider the propositional case and the execution of $\mathbf{der}_{\mathrm{LE}}(\Sigma, \emptyset, \emptyset, \phi)$ where the premises Σ are given in (17). The function $\mathbf{pos}_{\mathrm{LE}}$ expands gradually the set B and finally $\langle \{Lp_0, Lp_1, \ldots, Lp_{n-1}\}, \emptyset\rangle$ is returned. As $\mathbf{pos}_{\mathrm{LE}}(\Sigma, \emptyset, \emptyset)$ returns $\langle \{Lp_0, Lp_1, \ldots, Lp_{n-1}\}, \emptyset\rangle$, backtracking is not needed at all and the return value of $\mathbf{der}_{\mathrm{LE}}(\Sigma, \emptyset, \emptyset, \phi)$ is that of

$$\mathbf{test}(\Sigma, \{Lp_0, Lp_1, \ldots, Lp_{n-1}\}, \phi).$$

(ii) Consider the premises

$$\Sigma = \{\neg Lp \rightarrow q, \neg Lq \rightarrow p\}$$

and the execution of $\mathbf{der}_{\mathrm{LE}}(\Sigma, \emptyset, \emptyset, \phi)$. Here then $\mathbf{pos}_{\mathrm{LE}}(\Sigma, \emptyset, \emptyset)$ returns $\langle \emptyset, \emptyset\rangle$. As Lp and Lq are not covered by $\emptyset$ and p and q have no $L\chi$ subformulae, we can choose

FUNCTION $\mathbf{pos}_{\mathrm{LE}}(\Sigma, B, F)$
INPUT: Finite sets of sentences Σ, B, and F.

REPEAT
  $B' := B$;
  FOR every $L\chi \in \mathrm{Sf}^{\mathrm{L}}(\Sigma) - \text{q-atom}(B')$ **DO**
    IF $\Sigma \cup B \models_{\mathrm{ae}} \chi$ **THEN**
      IF $\mathrm{Sf}^{\mathrm{L}}(\chi) \subseteq \text{q-atom}(B)$ **THEN**
        $B := B \cup \{L\chi\}$
      ELSE
        $F := F \cup \{L\chi\}$
UNTIL $B = B'$;
return $\langle B, F \rangle$.

Figure 3: The Function $\mathbf{pos}_{\mathrm{LE}}$ for Positive Introspection Based on L-hierarchic Enumerations

either of them. Let us choose Lp. Hence $\mathbf{der}_{\mathrm{LE}}(\Sigma, \{\neg Lp\}, \emptyset, \phi)$ is executed next. Now $\mathbf{pos}_{\mathrm{LE}}(\Sigma, \{\neg Lp\}, \emptyset)$ returns $\langle \{\neg Lp, Lq\}, \emptyset \rangle$ and

$$\mathbf{test}(\Sigma, \{\neg Lp, Lq\}, \phi)$$

is executed. If this test returns true, $\mathbf{der}_{\mathrm{LE}}(\Sigma, \emptyset, \emptyset, \phi)$ returns true. Otherwise $\mathbf{der}_{\mathrm{LE}}(\Sigma, \emptyset, \{Lp\}, \phi)$ is executed. Because $\mathbf{pos}_{\mathrm{LE}}(\Sigma, \emptyset, \{Lp\})$ returns $\langle \emptyset, \{Lp\} \rangle$, $\mathbf{der}_{\mathrm{LE}}(\Sigma, \{\neg Lq\}, \{Lp\}, \phi)$ is executed. Since now $\mathbf{pos}_{\mathrm{LE}}(\Sigma, \{\neg Lq\}, \{Lp\})$ returns $\langle \{\neg Lq, Lp\}, \{Lp\} \rangle$,

$$\mathbf{test}(\Sigma, \{\neg Lq, Lp\}, \phi)$$

is executed. If this returns true, then $\mathbf{der}_{\mathrm{LE}}(\Sigma, \emptyset, \emptyset, \phi)$ returns true. Otherwise $\mathbf{der}_{\mathrm{LE}}(\Sigma, \emptyset, \{Lp, Lq\}, \phi)$ is executed and because $\mathbf{pos}_{\mathrm{LE}}(\Sigma, \emptyset, \{Lp, Lq\})$ returns $\langle \emptyset, \{Lp, Lq\} \rangle$, $\mathbf{der}_{\mathrm{LE}}(\Sigma, \emptyset, \emptyset, \phi)$ returns false. ■

If the underlying consequence relation $\models_{\mathrm{ae}}$ is decidable, decision procedures for cautious reasoning, brave reasoning, and the L-hierarchic expansion existence problem can be obtained by employing an appropriate function **test** and taking the empty set as the initial common part of the full sets. These **test** functions are straightforward to devise because membership in an expansion is captured by the consequence relation $\models_{\mathrm{L}}$, i.e. given a Σ-full set Λ, ϕ is in the corresponding unique expansion if and only if $\Sigma \cup \Lambda \models_{\mathrm{L}} \phi$.

Theorem 5.14 (Niemelä [91]) *Let $\Sigma \subseteq \mathcal{L}_{\mathrm{L}}$ be a finite set of sentences and $\phi \in \mathcal{L}_{\mathrm{L}}$ a sentence.*

1. If $\mathbf{test}(\Sigma, \Lambda, \phi)$ *returns true for any* Σ, Λ, ϕ, *then* $\mathbf{der}_{\mathrm{LE}}(\Sigma, \emptyset, \emptyset, \phi)$ *returns true if and only if* Σ *has an L-hierarchic expansion.*

2. If $\mathbf{test}(\Sigma, \Lambda, \phi)$ *returns true if and only if* $\Sigma \cup \Lambda \models_{\mathrm{L}} \phi$, *then* $\mathbf{der}_{\mathrm{LE}}(\Sigma, \emptyset, \emptyset, \phi)$ *returns true if and only if there is an L-hierarchic expansion of* Σ *containing* ϕ.

3. If $\mathbf{test}(\Sigma, \Lambda, \phi)$ *returns true if and only if* $\Sigma \cup \Lambda \not\models_{\mathrm{L}} \phi$, *then* $\mathbf{der}_{\mathrm{LE}}(\Sigma, \emptyset, \emptyset, \phi)$ *returns true if and only if there is an L-hierarchic expansion of* Σ *not containing* ϕ.

The decision method described above is completely general. It does not rely on any additional assumptions about the underlying logic which determines the consequence relation $\models_{\mathrm{ae}}$. For example, no normal form transformation of the premises is required. The method needs a theorem prover for $\models_{\mathrm{ae}}$ as a subroutine but no extra requirements are posed on the theorem prover.

Properties of the underlying logic as well as possible regularities of the sets of premises can be utilized to optimize the computationally costly subtasks in the decision method: required monotonic reasoning and conflict resolution. The theorem prover for $\models_{\mathrm{ae}}$ can exploit any special properties of the underlying logic or restricted form of the premises to achieve highest possible performance. An interesting topic of further research is to study possibilities of exploiting the regular pattern in which the theorem prover for $\models_{\mathrm{ae}}$ is used and make use of previous computations when deciding $\models_{\mathrm{ae}}$.

Optimization of conflict resolution is also a challenging area of further research. An interesting approach is to investigate the notion of safe disbeliefs. The idea is to identify disbeliefs $\neg L\chi$ such that χ does not follow from Σ and any possible extension of $B \cup \{\neg L\chi\} \cup F$. This reduces backtracking as then B can be extended safely by $\neg L\chi$ since for $\neg L\chi$ a conflict is not possible.

5.1.5 A Decision Method for Default Logic

As an example of developing optimization techniques we present a new complete decision method for Reiter's default logic [91]. By Theorem 5.4 default logic can be seen as a special case of autoepistemic logic based on L-hierarchic expansions. The simple logical character of the $L\chi$ sentences can be exploited when deciding $\models_{\mathrm{ae}}$ and the set of premises can be reduced considerably when the premises are restricted to the form resulting from the translation (9).

Proposition 5.15 (Niemelä [89]) *Let the underlying logic be a first-order calculus. Let $\Sigma \subseteq \mathcal{L}_{\mathrm{L}}$ be a set of sentences of the form $(Lb_1 \wedge \cdots \wedge Lb_n) \rightarrow c$ where $c \in \mathcal{L}$ and $n \geq 0$ and Λ is a consistent set of $L\psi$ sentences and their negations. Then for all $\phi \in \mathcal{L}$,*

$$\Sigma \cup \Lambda \models_{\mathrm{ae}} \phi \text{ if and only if } \mathrm{Hds}(\Sigma, \Lambda) \models \phi$$

where $\mathrm{Hds}(\Sigma, \Lambda) = \{c \mid (Lb_1 \wedge \cdots \wedge Lb_n) \rightarrow c \in \Sigma \text{ and } Lb_i \in \Lambda \text{ for all } i = 1, \ldots, n\}$.

An interesting notion of safe disbeliefs can be developed for the subclass of autoepistemic sentences resulting from the translation (9). The idea is to strengthen the premises Σ by "assuming" $\neg L\phi$ for every $L\phi \notin B \cup F$ in addition to those $\neg L\phi$ already in B. It turns out that a disbelief $\neg L\psi$ is safe if ψ does not follow from the strengthened premises Σ^* and B using positive introspection. The function $\mathbf{safeneg}_{\mathrm{LE}}$ in Figure 4 computes these safe disbeliefs.

It is easy to exploit safe disbeliefs in the decision method. Instead of propagating positive introspection by using $\mathbf{pos}_{\mathrm{LE}}$, both positive and safe negative introspection are propagated. This means that the function $\mathbf{intro}_{\mathrm{LE}}$ in Figure 4 is used instead of

FUNCTION $\mathbf{safeneg}_{\mathrm{LE}}(\Sigma, B, F)$
INPUT: Finite sets of sentences Σ, B, and F.

$\Sigma^* := \{La \rightarrow c \mid La \wedge L\neg Lb_1 \wedge \cdots \wedge L\neg Lb_n \rightarrow c \in \Sigma,$
for all $i = 1, \ldots, n, \neg Lb_i \in B$ or $Lb_i \notin B \cup F\}$;
$\langle B^*, F^* \rangle := \mathbf{pos}_{\mathrm{LE}}(\Sigma^*, B, F)$;
return $\{\neg L\psi \in \neg(\mathrm{Sf}^{\mathrm{L}}(\Sigma) - \text{q-atom}(B)) \mid \psi \in \mathcal{L}, \Sigma^* \cup B^* \not\models_{\mathrm{ae}} \psi\}$.

FUNCTION $\mathbf{intro}_{\mathrm{LE}}(\Sigma, B, F)$
INPUT: Finite sets of sentences Σ, B, and F.

REPEAT
$B' := B$;
$\langle B, F \rangle := \mathbf{pos}_{\mathrm{LE}}(\Sigma, B, F)$;
$B := B \cup \mathbf{safeneg}_{\mathrm{LE}}(\Sigma, B, F)$;
IF $\Sigma \cup B$ is consistent and
there exists $\neg L\chi \in B$ **THEN**
$B := B \cup \{\neg L\neg L\psi \in \neg(\mathrm{Sf}^{\mathrm{L}}(\Sigma) - \text{q-atom}(B)) \mid L\psi \in B\}$
UNTIL $B = B'$;
return $\langle B, F \rangle$.

Figure 4: Functions for Safe Negative Introspection and Propagation of Introspection

$\mathbf{pos}_{\mathrm{LE}}$ in the beginning of $\mathbf{der}_{\mathrm{LE}}$. Notice that $\mathbf{intro}_{\mathrm{LE}}$ also computes safe disbeliefs which are based on the fact that $\neg L\neg L\psi$ belongs to every consistent L-hierarchic expansion containing $L\psi$. We call the resulting function $\mathbf{der}_{\mathrm{DL}}$. The function $\mathbf{der}_{\mathrm{DL}}$ is sound and complete when given premises of the form resulting from the translation (9) [91].

Example 5 Consider the propositional case, the premises

$$\Sigma = \{L\neg Lp \rightarrow q, L\neg Lq \rightarrow p\}$$

and the execution of $\mathbf{der}_{\mathrm{DL}}(\Sigma, \emptyset, \emptyset, \phi)$. Compare this to Example 4 where safe negative introspection is not used. First $\mathbf{intro}_{\mathrm{LE}}(\Sigma, \emptyset, \emptyset)$ returns $\langle \emptyset, \emptyset \rangle$. Here we can extend the full set by $(\neg)Lp$ or $(\neg)Lq$. We choose Lp. Hence first $\mathbf{der}_{\mathrm{DL}}(\Sigma, \{\neg Lp\}, \emptyset, \phi)$ is executed. As $\langle \{\neg Lp, L\neg Lp, Lq, \neg L\neg Lq\}, \emptyset \rangle$ is returned by $\mathbf{intro}_{\mathrm{LE}}(\Sigma, \{\neg Lp\}, \emptyset)$,

$$\mathbf{test}(\Sigma, \{\neg Lp, L\neg Lp, Lq, \neg L\neg Lq\}, \phi)$$

is executed. If this test returns true, $\mathbf{der}_{\mathrm{DL}}(\Sigma, \emptyset, \emptyset, \phi)$ returns true. Otherwise $\mathbf{der}_{\mathrm{DL}}(\Sigma, \emptyset, \{Lp\}, \phi)$ is executed. As $\langle \{Lp, \neg L\neg Lp, \neg Lq, L\neg Lq\}, \emptyset \rangle$ is returned by $\mathbf{intro}_{\mathrm{LE}}(\Sigma, \emptyset, \{Lp\})$, $\mathbf{der}_{\mathrm{DL}}(\Sigma, \emptyset, \emptyset, \phi)$ returns what

$$\mathbf{test}(\Sigma, \{Lp, \neg L\neg Lp, \neg Lq, L\neg Lq\}, \phi)$$

returns. ■

5.2 Default Logic

The semantics of a default theory is given by the notion of extension. An extension is a set of formulae which can be obtained by applying default rules to the set of general laws. Normally, a default theory can have more than one extension, due to contradictions between consequences and between consequences and justifications. Two concepts of derivabilty are then possible:

- in the cautious (or sceptical) approach, only facts which occur in all extensions are believed. In this case, a formula is derivable when it belongs to every extension.
- in the credulous approach, all facts occourring in any extension are possibly accepted. In this case, a formula is derivable when it belongs to some extension.

Sometimes a default theory does not admit any extension. The most simple (canonical) example for a default theory with no extension is:

Example 6 $\Delta_6 = (W_6, D_6)$, with $W = \emptyset$ and $D = \dfrac{: \neg A}{A}$ ■

There is an important class of defaults, called *normal* defaults which always admit extensions. A normal default is a default with one justification which is identical to the consequent.

Reiter in his original article [102] gives an algorithm for computing extensions of a normal default theory without free variables. He defines default proofs in the following way:

A formula F has a default proof with respect to Δ iff there exists a finite sequence $D_0, D_1, \ldots D_k$ of subsets of D such that

1. $W \cup CONS(D_0) \vdash F$
2. For $1 \leq i \leq k, W \cup CONS(D_i) \vdash PREREQ(D_{i-1})$
3. $D_k = \emptyset$
4. $W \cup \bigcup_{i=0}^{k} CONS(D_i)$

Reiter [102] shows that a formula F has a default proof with respect to a default theory Δ if and only if F is contained in an extension of Δ. This notion of derivability is a cautious approach. Note, that the extension membership problem is not even semi-decidable, since 4 is not effectively enumerable. Reiter's definition of a proof for a default theory is incremental. It suggests the construction of a proof by applying successively applicable defaults and then to add the default consequences.

One of the first proof systems for default logic is due to Besnard et al. [8]. This system is restricted to propositional defaults without preconditions.

Schwind and Risch propose a proof system, based on the use of analytic tableaux [109, 111]. Schwind gives a first version of this system for normal default theories defined on a decidable subset of first order logic [109]. Schwind and Risch extend this work to default theories in the general case [111]. We will present this approach here in more detail.

5.2.1 Characterization of extensions

An extension of a default theory (W, D) is obtained by adding to the base set W default consequences.

The set of generating defaults of an extension E of a default theory without free variables $\Delta = (W, D)$ is defined by $GD(E, D) = \{d : d \in D, d = \frac{\alpha : \{\beta_1, \ldots, \beta_n\}}{\gamma}, \alpha \in E \text{ and } \neg\beta \notin E\}$

The set of consequents of a set of defaults D is defined by

$$CONS(D) = \{\gamma : \frac{\alpha : \{\beta_1, \ldots, \beta_n\}}{\gamma} \in D\}$$

The set of justifications of a default $d = \frac{\alpha : \{\beta_1, \ldots, \beta_n\}}{\gamma}$ is

$JUST(d) = \{\beta_1, \beta_2, ...\beta_n\}$, and the set of justifications of a set of defaults is the union of all the justification sets.

Given a set of formulae, F, we denote by $\neg F = \{\neg f : f \in F\}$ the set of negations of the elements of F.

The following theorem, which characterizes extensions in terms of theories over W and default consequences, has been proven in [102]:

Theorem 5.16 *If E is an extension for a default theory without free variables $\Delta = (W, D)$, then $E = Th(W \cup CONS(GD(E, \Delta)))$*

Theorem 5.16 shows that *if* a default theory (W, D) has and extension, then this extension is the set of theorems over W and a set of default consequences. But theorem 5.16 does not assert anything for default theories which do not have extensions. Hence, theorem 5.16 is of no help when given an arbitrary nonnormal default theory unless it is not known that it admits an extension. In [110], Risch and Schwind give a necessary *and* sufficient existence criterion for extensions. This result is based on the notion of *groundedness* of default sets.

Definition 5.17 (Schwind 90) *A set of defaults without free variables is grounded in W iff for all $d \in D$ there is a finite sequence d_0, d_1,...d_n of elements of D, $d_i = \frac{\alpha_i : \beta_1^i \cdots \beta_{n_i}^i}{\gamma_i}$ such that $\alpha_0 \in Th(W)$,$\alpha_{i+1} \in Th(W \cup CONS(\{d_0, d_1, ...d_i\}))$ for $0 \leq i \leq n-1$ and $d_n = d$.*

The following theorem has been proven by Schwind and Risch in [110].

Theorem 5.18 *A default theory $\Delta = (W, D)$ has an extension, E, iff there exists $D' \subseteq D$, D' grounded in W and $E = Th(W \cup CONS(D'))$ and $\forall\, d = \frac{\alpha : \beta}{\gamma}$ (i) and (ii) hold:*

(i) if $d \in D'$ then $\alpha \in Th(W \cup CONS(D'))$ and $\neg\beta_i \notin Th(W \cup CONS(D'))$, for all i such that $1 \leq i \leq n$;

(ii) if $d \notin D'$ then $\alpha \notin Th(W \cup CONS(D'))$ or $\neg\beta_i \in Th(W \cup CONS(D'))$ for some i such that $1 \leq i \leq n$.

5.2.2 Computing extensions

Several proof systems for default logic have been proposed since its definition by Reiter. Reiter in his default paper [102] proposes a deductive system for normal default theories. His system is based on resolution and decides if a given formula is in an extension. Besnard, Quiniou and Quinton [8] propose a proof procedure for normal defaults without prerequisites. TMS have been used by several authors for default logic proof systems [44, 57, 43]. Schwind [109] and Schwind and Risch [111]

propose to use analytic tableaux for the generation of extensions. We will present this approach more in detail.

The idea behind Risch and Schwind's method is very simple. An extension E for a default theory $\Delta = (W, D)$ is, roughly speaking, the set of theorems over the union of W and a set of default consequences (see theorem 5.16). This set exists whenever the conditions (i) and (ii) hold (see theorem 5.18). It is a maximal set of this kind in the sense that the addition of any other default consequence is not possible because the prerequisite or the justification condition does not hold or this would make E inconsistent (and then falsify the second part of condition (i)). Theorem 5.18 suggests that all subsets D' of D be found which correspond to extensions: if it is not possible to find such a subset (this is because of the fact that the justification condition can never be satisfied), then the theory has no extension.

Starting with the set $W \cup CONS(D)$, the algorithm for computing the extensions proceeds in three steps.

1. Consider the maximal consistent subsets of $W \cup CONS(D)$ which contain W. In other words, consider a collection of subsets D_1, ..., D_n of D, such that $W \cup CONS(D_i)$ is maximal consistent for $1 \leq i \leq n$. This means that, for every D_i and every $d \in D \setminus D_i$, $W \cup CONS(D_i \cup \{d\})$ is inconsistent. Each of the D_i is a candidate to be a set of generating defaults of an extension.

2. The justification conditions are tested for every extension candidate $Th(W \cup CONS(D_i))$. This yields maximal subsets D_i^j of D_i such that $Th(W \cup CONS(D_i^j)) \cap \neg JUST(D_i^j)) = \emptyset$.

3. From the $Th(W \cup CONS(D_i^j))$, the grounded sets are obtained by eliminating within D_i^j defaults which are not grounded. Testing groundedness consists in verifying for every default $d = \frac{\alpha : \{\beta_1, \ldots, \beta_n\}}{\gamma}$ of D_i^j that $\alpha \in Th(W \cup CONS(D_i^j \setminus \{d\}))$: if $\alpha \in Th(W)$, then $\{d\}$ is grounded. If $\alpha \in CONS(D_i^j \setminus \{d\})$ then $D_i^j \setminus \{d\}$ has to be grounded. A sequence of verifications is thereby generated for some defaults in D_i^j in such a way that each default used for a verification is dropped out of D_i^j. D_i^j is grounded if and only if at least one prerequisite of the sequence belongs to W, i.e. dropping defaults of the sequence after each verification does not lead to the empty sequence. Note that it is not more costly to test the groundedness of D_i^j than to test the prerequisite condition on the defaults of D_i^j.

4. Lastly, it has to be verified that extensions are minimal and do not contain each other.

The first step in this algorithm treats a special case of the second step, since in the case of inconsistency, no justification condition is verified, because an inconsistent set contains every formula.

Schwind and Risch's system uses a prover, called TP which is based on the analytic tableaux method in the style of [114] and is described in [109]. TP proves a formula

F by trying to derive a contradiction from $\neg F$. Let us consider the usual Smullyan classes of formulae [26, 114]:

α	α_1	α_2	β	β_1	β_2
$f \wedge g$	f	g	$f \vee g$	f	g
$\neg(f \vee g)$	$\neg f$	$\neg g$	$\neg(f \wedge g)$	$\neg f$	$\neg g$
$\neg(f \rightarrow g)$	f	$\neg g$	$f \rightarrow g$	$\neg f$	g

γ	$\gamma(p), p$ any parameter	δ	$\delta(p), p$ Skolem parameter
$(\forall x)f$	$f[x/p]$	$(\exists x)f$	$f[x/p]$
$\neg(\exists x)f$	$\neg f[x/p]$	$\neg(\forall x)f$	$\neg f[x/p]$

where parameters are free occurrences of variables or functions whose arguments are parameters and Skolem parameters are in the form $\varphi(p_1, \ldots, p_n)$, where φ is any skolem function, and $p_1, \ldots, p_n$ are *already ground* parameters. TP can be defined recursively as a mapping between sets of formulas and sets of literals as follows:

$$
\begin{aligned}
TP(F) &= \{F\} \text{ if } F \text{ is a set of literals;} \\
TP(F) &= TP(F' \cup \{f\} \text{ if } \neg\neg f \in F \text{ and } F' = F \setminus \{\neg\neg f\}; \\
TP(F) &= TP(F' \cup \{\alpha_1\} \cup \{\alpha_2\}) \text{ if } \alpha \in F \text{ and } F' = F \setminus \{\alpha\}; \\
TP(F) &= TP(F' \cup \{\beta_1\}) \cup TP(F' \cup \{\beta_2\}) \text{ if } \beta \in F \text{ and } F' = F \setminus \{\beta\}; \\
TP(F) &= TP(F' \cup \{\gamma(p)\} \text{ if } \gamma \in F \text{ and } F' = F \setminus \{\gamma\}; \\
TP(F) &= TP(F' \cup \{\delta(p)\} \text{ if } \delta \in F \text{ and } F' = F \setminus \{\delta\};
\end{aligned}
$$

Note that the analytic tableau for a formula corresponds to its disjunctive normal form.

Definition 5.19 *A set of literals is* closed[3] *if it contains two opposite literals (i.e. ℓ and $\neg\ell$). It is* open *otherwise. A set of sets of literals is closed if each of its elements is closed, and open otherwise.*

The fundamental property of TP for theorem proving is the following completeness theorem:

Theorem 5.20 (Smullyan, 68) *f is a theorem iff $TP(\{\neg f\})$ is closed.*

We consider TP not only as a theorem prover (or consistency checker) for first-order formulas but also as an application which has useful properties for formulas and formulas sets. We will frequently use the following properties of TP:

Lemma 5.21 $TP(\{f \wedge g\}) = \{X \cup Y : X \in TP(\{f\}) \text{ and } Y \in TP(\{g\})\}$.

[3]not to be confused with a *closed* first-order formula i.e. a formula with no free variables.

This will enable us to consider conjunction as an operation upon sets of sets of literals. Lemma 5.21 gives rise to the following notation:

$$f \otimes g = \{X \cup Y : X \in TP(\{f\}) \text{ and } Y \in TP(\{g\})\}.$$

Thus $TP(\{f \wedge g\}) = TP(\{f\}) \otimes TP(\{g\})$. TP will only have to compute the image of each formulas f and g *once*, and the image of $TP(\{f \wedge g\})$ will be obtained using the operation $\otimes$ without having to compute the whole conjunct. As an immediate consequence of the completeness theorem, we have

Corollary 5.22 (Schwind, 90) *Let Γ be a finite set of formulas and f a formula. $f \in Th(\Gamma)$ iff $TP(\Gamma) \otimes TP(\{\neg f\})$ is closed.*

Remark: (1) TP frequently contains closed sets or sets which contain other sets in TP. In this case we are sometimes interested in a minimal form (especially when TP is not closed), defined by $TS(M) = TP(M) \setminus \{X : X \in TP(M)$ and X closed or $\exists Y \in TP(M)$ such that $Y \subseteq X$ and $X \neq Y\}$. Very often this can considerably shorten a tableau. (2) Determining the class of formulas that a tableau prover can decide is a difficult question which is beyond the scope of this paper. In what follows, we will consider only decidable subclasses of a first-order langage e.g. propositional formulas, Horn clauses, or the subclass of the formulas having no function symbols other than constants and no existential quantifiers.

The tableaux method is very interesting for the generation of extensions, mainly because of the fundamental property. The tableau for f is closed iff $\neg f$ is a theorem, i.e. a tableau for f is open iff $\neg f$ is not a theorem, which means that f is consistent. Nonmonotonic reasoning typically requires reasoning about consistency versus inconsistency. A tableau provides a straightforward means of transforming inconsistent sets into consistent ones by *removing* the literals responsible for contradiction within these sets. Similarly, the $\otimes$ operator appears to be very tractable, since using the fundamental property of TP, and given a default $d = \dfrac{\alpha : \beta}{\gamma}$, we obtain (cf. theorem 5.18):

(i) If $d \in D'$ then $TP(\{\neg\alpha\}) \otimes TS(W) \otimes TP(CONS(D'))$ is closed and $TP(\{\beta\} \otimes TS(W) \otimes TP(CONS(D'))$ is open;

(ii) If $d \notin D'$ then $TP(\{\neg\alpha\} \otimes TS(W) \otimes TP(CONS(D'))$ is open or $TP(\{\beta\} \otimes TS(W) \otimes TP(CONS(D'))$ is closed;

where D' is the set of defaults involved in the constitution of the extension $Th(W \cup CONS(D'))$. Note that *a priori* we are not allowed to use TS in preference to TP with $CONS(D')$, since otherwise some defaults could be forgotten by the process (e.g. the consequent of a given default is subsumed by the consequent of an other default which cannot actually be used).

The groundedness of each default involved in an extension $Th(W \cup CONS(D'))$ is verified in a recursive way: Choose $d \in D'$, $d = \dfrac{\alpha : \beta}{\gamma}$. If $TP(W \otimes CONS(D' \setminus$

$\{d\}))\otimes\{\neg\alpha\})$ is closed, a finite sequence $d_0,\ldots,d_n$ of elements of D' has to be found, $d_i = \dfrac{\alpha_i : \beta_i}{\gamma_i}$, such that $TP(\{\neg\alpha_0\})\otimes TP(W)$ is closed, $TP(\{\neg\alpha_{i+1}\})\otimes TP(W)\otimes TP(CONS(\{d_0,\ldots,d_i\})$ is closed for $0{\leq}i{\leq}n-1$ and $d_n = d$. However, if there exists k, $0{\leq}k{\leq}n$, such that $TP(\{\neg\alpha_k\})\otimes TP(W)\otimes TP(CONS(D'\setminus\{d_k,\ldots,d_n\}))$ is open i.e. $\alpha_k \notin Th(W\cup CONS(D'\setminus\{d_k,\ldots,d_n\}))$ the sequence $d_k,\ldots,d_n$ has to be withdrawn from D'. In this way all grounded extensions are obtained.

Example 7 $\Delta = (W,D)$ where $W=\{A,F\}$, $D=\{\dfrac{A : B\wedge\neg C}{B}, \dfrac{F : C}{C}\}$. $TS(W) = \{\{A,F\}\}$, $TP(CONS(D)) = \{\{B_1,C_2\}\}$ where B_1 and B_2 are the consequents of the first and the second default respectively. Consider D as a possible candidate in order to obtain an extension. However, the justification condition of *(i)* is not true for the first default: $TS(W)\otimes TP(D)\otimes TP(B_1\wedge\neg C_1) = \{\{A,F,B_1,C_2,\neg C_1\}\}$ is closed (where $\neg C_1$ comes from the justification of the first default). Hence, consider the following subsets of D: $D' = \{\dfrac{A : B\wedge\neg C}{B}\}$ and $D'' = \{\dfrac{F : C}{C}\}$. Now, both D' and D'' verify the justification condition of *(i)*: $TS(W)\otimes TP(D')\otimes TP(\{B_1\wedge\neg C_1\}) = \{\{A,F,B_1,\neg C_1\}\}$ is open and $TS(W)\otimes TP(D')\otimes TP(\{C_2\}) = \{\{A,F,C_2\}\}$ is open. But E_1 does not verify the second condition *(ii)* since for the second default $\dfrac{F : C}{C}$ $F\in E_1$ and $\neg C\neg\in E_1$: $TS(W)\otimes TP(\{B_1\})\otimes TP(\{C_2\})$ is open instead of being closed (where C_2 is the justification of the second default). Hence E_1 is not an extension. E_2 is a possible extension since $TS(W)\otimes TP(\{C_2\})\otimes TP(\{B_1\wedge\neg C_1\})$ is closed, which is expected following *(ii)*.E_2 verifies both justification conditions (i) and (ii) and is obviously grounded. Thus D has one extension:

$$E_2 = Th(W_1\cup CONS(\{\frac{F : C}{C}\})).$$

■

Example 8 $\Delta = (W,D)$ where $W=\{P,S\}$, $D=\{\dfrac{: Q\wedge S}{P}, \dfrac{: Q\rightarrow\neg S}{\neg R}\}$. $TS(W) = \{\{P,S\}\}$, $TP(CONS(D)) = \{\{P_1,\neg R_2\}\}$. First consider D as the possible unique candidate in order to get an extension. The justification condition of *(i)* holds for both of the defaults:

- $TS(W)\otimes TP(CONS(D)\otimes TP(\{Q_1\wedge S_1\}) = \{\{P,S,P_1,\neg R_2,Q_1,S_1\}\}$ is open.
- $TS(W)\otimes TP(CONS(D)\otimes TP(\{Q_2\rightarrow\neg S_2\})$
  $= \{\{P,S,P_1,\neg R_2,\neg Q_2\},\{P,S,P_1,\neg R_2,\neg S_2\}\}$ is open;

Since D is obviously grounded, Δ has a single extension

$$E = Th(W\cup CONS(\{\frac{: Q\wedge\neg S}{P}, \frac{: Q\rightarrow\neg S}{\neg R}\})).$$

■

Example 9 $\Delta = (\emptyset, D)$ where $D = \{\frac{A : B}{B}, \frac{B : A}{A}\}$. The justification condition holds for both defaults of D. However, D is not grounded, neither is $\{\frac{A : B}{B}\}$ or $\{\frac{B : A}{A}\}$. Hence D has only one extension $E = \emptyset$. ■

In the algorithm below the formulas are indexed by the number of each default, as in the previous examples. The variable D_{just} represents the set of defaults that satisfy the justification condition, the variable D_{grnd} the set of grounded defaults, and the variable SD_{just} the set of the D_{just}. For any subset D' of D, the function $\Gamma(D')$ (used inside the procedure `JUSTIFICATIONS` below) returns the set $TP(W) \otimes TP(CONS(D'))$. The part of the algorithm that tests the grounding of a set of defaults uses a variable S that represents a grounding path.

```
COMPUTE-EXTENSIONS(W, D):
    If W is not consistent
        then PRINT(W is not consistent)
        else For any maximal subset D' in D such that TP(W ⊗ CONS(D')) is open do
            D_just := ∅; D_grnd := ∅; SD_just := {D_just};
            JUSTIFICATIONS(D', D_just, SD_just);
            For any D_just in SD_just do
                GROUNDING(D_just, ∅, D_grnd);
                EXTENSIONS(D_just, D_grnd).

JUSTIFICATIONS(D', D_just, SD_just):
    For any default d = (α : β)/γ ∈ D' do
        If TP({β}) ⊗ Γ(D') is open
            then Add d to D_just
            else For any maximal subset D'' in D' such that TP({β}) ⊗ Γ(D'') is open, do
                JUSTIFICATIONS(D'', D_just, SD_just);
                If (D_just ∉ SD_just) and (there is no D'_just ∈ SD_just such that D_just ⊆ D'_just)
                    then Add D_just to SD_just

GROUNDING(D', S, D_grnd):
    For any default d = (α : β)/γ ∈ D' \ D_grnd do
        Θ := Th(W ⊗ CONS(D' \ (S ∪ {d})));
        If TP(¬α) ⊗ Θ is closed,
            then If α ∈ Th(W) or comes from a grounded set of defaults
                then Add d to D_grnd
                else Let be D_Θ the set of defaults of D' \ (S ∪ {d})
                whose the consequent is used for deriving α, do
                    GROUNDING(D_Θ, S ∪ {d});
                    If D_Θ is grounded then Add d to D_grnd;
            else Remove the list of defaults S ∪ {d}
            from D'.

EXTENSIONS(D_just, D_grnd):
    If D_grnd is a maximal grounded subset of D then
        PRINT(Th(W ∪ CONS(D_grnd)) is a j-extension with respect to JUST(D_grnd)).
```

Remark: It should be stressed that any check already done for a justication regarding some set D' of defaults still holds for a subset of D': if the justification condition holds with respect to D', no contradiction can be found in a subset of D' and the check should not be repeated. If the justification condition does not hold, the set D' is broken in such a way that the cause of the contradiction is removed, and hence the check cannot be repeated anyway. This remark allows us to check the justification condition of each default only once, provided that we remember which defaults have been already treated. In the same way prerequisite condition for each default is checked only once.

5.3 Circumscription

Predicate circumscription gives a syntactical account of predicate minimization. Consider the sentence

$$Bird(Tweety) \wedge Snake(Kaa).$$

Without applying circumscription to minimize the predicate $Bird$, it would be extremely difficult to deduce that Kaa is not a bird.

Circumscribing $Bird$ in the above example gives $Circ(Bird(Tweety) \wedge Snake(Kaa); Bird) = Bird(Tweety) \wedge Snake(Kaa) \wedge \neg\exists\Phi(\Phi(Tweety) \wedge Snake(Kaa) \wedge \Phi < Bird)$. Instantiating Φ with $x = Tweety$ allows us to deduce $\neg Bird(Kaa)$ (using the unique names assumption $Tweety \neq Kaa$).

Now consider the following sentences expressing that "birds typically fly". This should be formalizable using circumscription in order to minimize exceptions.

$$Bird(Tweety)$$

$$\forall x((Penguin(x) \rightarrow Bird(x) \wedge \neg fly(x)).$$

$$\forall x((Bird(x) \wedge \neg exception(x)) \rightarrow fly(x)).$$

Tweety is a bird, penguins are birds which do not fly and a bird flies unless it is an exceptional bird. By circumscribing the predicate *exceptional*, it should be possible to minimize exceptions to only those which are known to be exceptional.

But circumscribing the predicate *exception* in the example does not permit to derive $fly(Tweety)$. This is due to the fact that there is a dependency between *exception* and fly which cannot be taken into account. This limitation of predicate circumscription is remedied by generalizing predicate circumscription to formula circumscription [79]. A formula is circumscribed with a set of predicate and function constants which are allowed to vary.

Circumscribing *exception* in the example with fly allowed to vary then makes it possible to derive $fly(Tweety)$.

Actions are generally characterized by two formulae, representing the preconditions P and the results R of the action. Action laws then express that an action can be perfored provided P holds and as a result R will hold. But this still does not make it possible to deduce that *only* R constitutes the change caused by the action performance. Most facts will *not* change as an action occurs. Circumscription has been used to provide a solution to the frame problem.

The laws

$$Holds(P, s^i) \rightarrow Holds(R, s^r)$$

$$\neg ab(f) \rightarrow (Holds(f, s^i) \leftrightarrow Holds(f, s^r))$$

formalize the performance of the action, where s^i and s^r design the initial and the resulting states. The *frame axiom* expresses that a fact f does not change as an action occurs unless it is abnormal. Abnormality is then minimized by circumscribing the predicate ab with $Holds$ allowed to vary.

Proof systems for circumscription encounter several severe problems. Predicate as well as first-order formula circumscription is incomplete for universal finitary theories. Minimal models exist only for well-founded theories [7]. Circumscription is complete for the class of theories with a domain closure axiom. The presence of a second order formula in a first-order set of formulae in principle rules out a complete deductive system. But it is possible to distinguish classes of formulae for which circumscription is equivalent to a first-order formula. In section 5.3.1 we present these classes.

Another approach to the deduction problem is to generate the minimal models for the circumscription formula. We present a more recent developpment about minimal model generation in presence of fixed and varying predicates in section 5.3.2.

Olivetti's algorithm [93] for minimal entailment will be explained in section 5.3.3.

5.3.1 Expressing circumscription by first-order formulae

Since in general complete deductive systems for circumscription do not exist, it is fruitful to identify classes of formulae for which the circumscription is equivalent to a first-order formula. Lifschitz first [59] studied classes of formulae which admit a first-order circumscription. The following examples from [59] illustrate these classes.

Example 10 Let be $A = P(a)$. $Circ(A;P) \equiv \forall x(P(x) \leftrightarrow x = a)$ ■

Example 11 Let be $A = \neg P(a)$. $Circ(A;P) \equiv \forall x \neg P(x)$ ■

Example 12 Let be $A = \forall x(French(x) \rightarrow eat-cheese(x))$. Then $Circ(A; eat-cheese) \equiv \forall x(French(x) \leftrightarrow eat-cheese(x))$ ■

Minimizing eat-cheese in "every French eats cheese" results in "every cheeze eater is French.

The examples suggest several classes of formulae for which the result of circumscription can be first-order defined. In example 11, a formula without positive occurrence of the circumscribed predicates is circumscribed. In example 12, the circumscribed formula has two separate parts, one not containing circumscribed predicates and the other containing only one circumscribed predicate. These properties are generalized in the following definition (see [59]).

Definition 5.23 *Let P be a tuple of predicates P_1, P_2, ... P_n. $A(P)$ is said to be solitary with respect to P if $A(P)$ is a conjunction of*

(i) formulae with no positive occurrence of P_1, P_2, ...P_n.

(ii) formulae of the form $U \leq P_i$, where U is a tuple of predicates not containing P_1, P_2, ...P_n.

Hence, a solitary formula has the form $N(P) \wedge (U \leq P)$, N not containing P positively and U not containing P_1, P_2, ...P_n.

Given a solitary formula $F = N(P) \wedge (U \leq P)$, the result of circumscribing P in F is a first-ordre formula:

$Circ(N(P) \wedge (U \leq P); P) \equiv N(P) \wedge (U = P)$

In order to handle disjunctions, the notion of separable formula is introduced. A formula constructed from subformulae of the form (i) and (ii) using conjunctions and disjunctions is called *separable*. Consequently, any separable formula is equivalent to a disjunction of solitary formulas and can be written in the form

$$\bigvee_i ((N_i(P) \wedge (U^i \leq P))$$

N_i does not contain positive occurrences of P_1, P_2, ...P_n and each U^i is a tuple not containing P_1, P_2, ...P_n. The following theorem gives the result for circumscription of separable formulae.

Theorem 5.24 *If $A(P)$ is equivalent to a separable formula then $Circ(A(P); P)$ is equivalent to*

$$\bigvee_i (D_i \wedge (U^i = P))$$

, where

$$D_i = N_i(U^i) \wedge \bigwedge_{i \neq j} \neg(N_j(U^j) \wedge (U^j < U^i))$$

In [50] a more general result is given concerning the circumscription of existential formulas.

Theorem 5.25 *Let $A(P)$ be an existential first-order formula of the form $\exists x B$, where $x = (x_1, x_2, ...x_n)$ and B is a quantifier-free formula. Then the circumscription of P in A, $Circ(A; P)$ is equivalent to a first-order formula.*

5.3.2 Minimal Models generation

Minimal models generation is always a possible solution to the deduction problem for circumscription [6, 54]. Kuhna [54] uses analytic tableaux (as introduced in 5.2.2) for the generation of $<_{P_Z}$-minimal models of a given theory A. The implicants produced by the operators TP the selection of minimal models for a predicate is not so straightforward in the presence of other fixed predicates. Consider the following sets of literals:

$$T_1 = \{S(a), S(b)\}, T_2 = \{P(a), S(a)\}$$

Suppose we want to select the models which are minimal for P. At a first glance T_2 seems to be smaller than T_1. But the models associated to T_1 and T_2 are not directly comparable under this representation since T_2 does not contain $S(b)$. Hence T_2

represents one model which contains $S(b)$ and another model which does not contain $S(b)$. By expanding T_2 to $T_2' = \{P(a), S(a), S(b)\}$ and $T_2'' = \{P(a), S(a), \neg S(b)\}$ an "equivalent" tableau is obtained (i.e. a tableau representing an equivalent formula). Then the models represented by T_1 are minimal but so are the models represented by T_2'' since they do not coincide with the models from T_1 not containing P.

By systematically completing branches it is possible to obtain all minimal models in this way. Kuhna shows also how to generate automatcally instances for the circumscription scheme allowing to perform nonmonotonic deductions.

5.3.3 Minimal Entailment and Minimal Models

Circumscription is semantically characterized by minimal models semantics. Olivetti [93] proposes a decision method for minimal derivability for the propositional case. He defines mnimal entailment as follows:

Definition 5.26 *A formula B is said to be* minimally entailed *by a formula A if B holds in all minimal models of A. This is noted $A \models^{<} B$.*

Olivetti [93] uses signed tableaux as presented originally in the literature [114]. He presents an elegant modification of the usual tableaux rules in order to handle minimal entailment. Given a propositional language L, a signed formula has the form Tf or Ff, where $f \in L$. Propositional valuations are extended to signed formulae. Given an interpretation I, and a formula f, $I \models Tf$ iff $I \models f$ and $I \models Ff$ iff $I \models \neg f$. Tableaux rules are modified accordingly.

α	α_1	α_2	β	β_1	β_2
$T(f \wedge g)$	$T(f)$	$T(g)$	$T(f \vee g)$	$T(f)$	$T(g)$
$F(\neg(f \wedge g))$	$F(\neg f)$	$F(\neg g)$	$F(\neg(f \vee g)$	$F(\neg f)$	$F(\neg g)$
$T(\neg(f \vee g))$	$T(\neg f)$	$T(\neg g)$	$T(\neg(f \wedge g)$	$T(\neg f)$	$T(\neg g)$
$F(f \vee g)$	$F(f)$	$F(g)$	$F(f \wedge g)$	$F(f)$	$F(g)$
$T(\neg(f \rightarrow g)$	$T(f)$	$T(\neg g)$	$T(f \rightarrow g)$	$T(\neg f)$	$T(g)$
$F(f \rightarrow g)$	$F(\neg f)$	$F(g)$	$F(\neg(f \rightarrow g)$	$F(f)$	$F(\neg g)$

From the above rules it can be seen that the formulae generated from a formula keep the same sign.

Tableaux are defined as trees as usual, where type α-rules extend branches and type β-rules split branches.

Given a branch B, $At(B) = \{Tp : p \textit{ is an atom and } Tp \in B\}$ is the subset of the atoms of B.

Completed branches are defined as Hintikka sets.

Definition 5.27 *A branch B of a tableau is* completed *if:*

- *for every $\alpha \in B$ of type α, α_1 and α_2 are in B,*

- *for every $\beta \in B$ of type β, β_1 or β_2 are in B.*

The usual definition of closed branches is slightly modified in order to take into account the fact that signs never change during the tableau construction.

Definition 5.28 *A branch B is* closed *if one of the following holds:*

- *for some formula f, $Tf \in B$ and $T\neg f \in B$ (T-closed),*
- *for some formula f, $Ff \in B$ and $F\neg f \in B$ (F-closed),*
- *for some formula f, $Tf \in B$ and $Ff \in B$.*

The following definition is the important definition which makes it possible to ignore branches which do not represent minimal models.

Definition 5.29 *A branch B is* ignorable *if one of the following conditions holds:*

- *B is completed and for some atom p, $F\neg p \in B$, but and $Tp \notin B$,*
- *there is another branch B' which is completed but not T-closed such that $At(B') \subset At(B)$.*

A tableau is closed of every non-ignorable branch is closed.

Minimal deravability is then defined as expected: A minimally derives B, noted $A \vdash^T B$, if every tableau that starts with TA and FB is closed.

Olivetti shows the correctness and the completeness of his construction.

Theorem 5.30 *Let A and B be propositional formulae. Then $A \vdash^T B$ iff $A \models^< B$*

5.4 Conditional Logics

Conditional Logics have been introduced in order represent sentence structures like "If it were the case that, ...". But it is also possible to interpret such a *counterfactual* in a slightly different way. The conditional clause may refer to, situations which are most *normal* according to the main clause. With this understanding, the above sentence may also be read "Normally, if it is the case that, ..". This direction has been exploitet by [18, 19].

The language of conditional logic is the language of propositional logic augmented by a new binary operator $\Longrightarrow$. The following minimal axiom system is included in all conditional logics.

Definition 5.31 (System CK) *The minimal normal conditional Logic CK is the smallest system which contains the classical propositional axioms, is closed under modus ponens and the following rules:*

RCEA from $\vdash \alpha \leftrightarrow \beta$ *derive* $\vdash (\alpha \Longrightarrow \gamma) \leftrightarrow (\beta \Longrightarrow \gamma)$

RCK from $\vdash \beta_1 \vee \beta_2 \ldots \beta_n \rightarrow \beta$ *derive* $\vdash ((\alpha \rightarrow \beta_1) \wedge (\alpha \rightarrow \beta_2) \wedge \ldots (\alpha \rightarrow \beta_n)) \leftrightarrow (\alpha \Longrightarrow \beta)$

Lewis [58] has used another system, V in order to capture revision. The system V has also been used by Degrande [18] for dealing with normality.

Definition 5.32 (System V) *The system V is the smallest system containing CK and the following axioms:*

ID $\alpha \Longrightarrow \alpha$

CSO $((\alpha \Longrightarrow \beta) \wedge (\beta \Longrightarrow \alpha)) \rightarrow ((\alpha \Longrightarrow \gamma) \leftrightarrow (\beta \Longrightarrow \gamma))$

CA $((\alpha \Longrightarrow \gamma) \wedge (\beta \Longrightarrow \gamma)) \rightarrow ((\alpha \vee \beta) \Longrightarrow \gamma)$

CV $((\alpha \Longrightarrow \gamma) \wedge (\alpha \Longrightarrow \neg\beta)) \rightarrow ((\alpha \wedge \beta) \Longrightarrow \gamma)$

The semantics of this system can be given as a <Kripke -type semantics by means of a selection function on worlds. Lewis introduced a sphere systems semantics. The theorem prover given by Lamarre [55] for the system V is based on sphere semantics.

Definition 5.33 (Sphere Semantics) *A sphere models is a triple* $M = (W, S, v)$ *such that* W *is a nonempty set of worlds and* v *is a valuation function.* S *is a function* $W \longrightarrow 2^{2^W}$ *associating to each world* w *of* W *a sphere system* $S(w)$ *such that:*

P1 $S(w)$ *is* nested. *If* $s_1 \in S(w)$ *and* $s_2 \in S(w)$ *then* $s_1 \subseteq s_2$ *or* $s_2 \subseteq s_1$.

P2 $S(w)$ *is closed under union*

P3 $S(w)$ *is closed under non-empty intersection.*

The sphere system $S(w)$ belonging to the world w represents the worlds which are *normal* for w. Intuitively the construction expresses that the more normal is a world with respect to w, the smaller is the sphere within $S(w)$ to which it belongs.

Definition 5.34 (Satisfiability) *Satisfiability in a sphere system is defined as usual for the classical operators. If* M *is a sphere model and* w *one of its worlds, then* $M, w \models (\alpha \Longrightarrow \beta)$ *iff*

1. $\alpha \rightarrow \beta$ *is true in all the worlds of the smallest sphere which contains at least one world* w' *satisfying* α *or*
2. *There is no world satisfying* α *in any sphere.*

Lamarre's theorem prover for conditionals is based on analytic tableaux. As excpected, to prove that a formula A is a theorem, it proves that no model exists for its negation. So given a formula the problem $\neg A$, the prover tries to construct one of its models. He considers a sort of "disjunctive normal form". By a number of rules which apply to the elements of the disjuncts a shere system is built beginning by the smallest one.

References

[1] A.B. Baker. Nonmonotonic reasoning in the framework of situation calculus. *Artificial Intelligence*, 49:5–23, 1991.

[2] A.B. Baker and M. Ginsberg. Temporal projection and explanation. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 906–911, Detroit, USA, August 1989. Morgan Kaufmann Publishers.

[3] J.L. Balcázar, I. Díaz, and J. Gabarró. *Structural Complexity I.* Springer-Verlag, Berlin, 1988.

[4] R. Ben-Eliyahu and R. Dechter. Default logic, propositional logic and constraints. In *Proceedings of the 9th National Conference on Artificial Intelligence*, pages 370–385. The MIT Press, July 1991.

[5] R. Ben-Eliyahu and R. Dechter. Propositional semantics for default logic. In *Working Notes of the 4th International Workshop on on Nonmonotonic Reasoning*, pages 13–27, Plymouth, Vermont, USA, May 1992.

[6] R. Ben-Eliyahu, R. Dechter. On Computing Minimal Models. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 2–8, 1993.

[7] P. Besnard. *An Introduction to Default Logic.* Springer-Verlag, Berlin, 1989.

[8] P. Besnard, R. Quiniou, and P. Quinton. A theorem-prover for a decidable subtset of default logic. In *Proceedings of the National Conference on Artificial Intelligence*, pages 27–30, 1983.

[9] N. Bidoit. Negation in rule-based database languages: a survey. *Theoretical Computer Science*, 78:3–83, 1991.

[10] N. Bidoit and C. Froidevaux. General logical databases and programs: Default logic semantics and stratification. *Information and Computation*, 91:15–54, 1991.

[11] N. Bidoit and C. Froidevaux. Negation by default and unstratified logic programs. *Theoretical Computer Science*, 78:85–112, 1991.

[12] P.A. Bonatti. Autoepistemic logics as a unifying framework for the semantics of logic programs. In *Proceedings of the 1992 Joint International Conference and Symposium on Logic Programming*, Washington, D.C., USA, November 1992.

[13] G. Brewka. *Nonmonotonic Reasoning: Logical Foundations of Commonsense.* Cambridge University Press, Cambridge, 1991.

[14] M. Cadoli and M. Lenzerini. The complexity of closed world reasoning and circumscription. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 550–555, Boston, MA, USA, July 1990. The MIT Press.

[15] M. Cadoli and M. Schaerf. A survey of complexity results for non-monotonic logics. *Journal of Logic Programming*, 17:127–160, 1993.

[16] B.F. Chellas. *Modal Logic: an Introduction.* Cambridge University Press, Cambridge, 1980.

[17] K.L. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 293–322. Plenum Press, New York, 1978.

[18] J.P. Delgrande. A First-Order Conditional Logic for Prototypical Prperties. Artificial Intelligence, no.36, pages 63-90, 1988.

[19] J.P. Delgrande. An Approach to Default Reasoning Based on First-Order Conditional Logic: Revised Report. Artificial Intelligence, no.36, pages 63-90, 1988.

[20] T. Eiter and G. Gottlob. Propositional circumscription and extended closed world reasoning are Π_2^p-complete. Technical Report CD-TR 91/20, Christian Doppler Labor für Expertensysteme, Institut für Informationssysteme, Technische Universität Wien, Vienna, Austria, May 1991.

[21] T. Eiter and G. Gottlob. The complexity of logic-based abduction. Technical Report CD-TR 92/35, Christian Doppler Labor für Expertensysteme, Institut für Informationssysteme, Technische Universität Wien, Vienna, Austria, July 1992.

[22] C. Elkan. A rational reconstruction of nonmonotonic truth maintenance systems. *Artificial Intelligence*, 43:219–234, 1990.

[23] D.W. Etherington. Formalizing nonmonotonic reasoning systems. *Artificial Intelligence*, 31:41–85, 1987.

[24] D.W. Etherington. Relating default logic and circumscription. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pages 489–494, Milan, Italy, August 1987. Morgan Kaufmann Publishers.

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

[26] M. Fitting. *First Order Logic and Automated Theorem Proving.* Texts and Monographs in Computer Science. Springer Verlag, 1990.

[27] Y. Fujiwara and Honiden S. Relating TMS to autoepistemic logic. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 1199–1205, Detroit, USA, August 1989. Morgan Kaufmann Publishers.

[28] M.R. Garey and D.S. Johnson. *Computers and Intractability.* W.H. Freeman and Company, San Francisco, 1979.

[29] M. Gelfond. On stratified autoepistemic theories. In *Proceedings of the 6th National Conference on Artificial Intelligence*, pages 207–211, Seattle, USA, July 1987. American Association for Artificial Intelligence.

[30] M. Gelfond. Autoepistemic logic and formalization of commonsense reasoning, preliminary report. In *Proceedings of the 2nd Workshop on Non-Monotonic Reasoning*, pages 176–186, Grassau, FRG, June 1988. Springer-Verlag.

[31] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proceedings of the 5th International Conference on Logic Programming*, pages 1070–1080, Seattle, USA, August 1988. The MIT Press.

[32] M. Gelfond and V. Lifschitz. Logic programs with classical negation. In *Proceedings of the 7th International Conference on Logic Programming*, pages 579–597, Jerusalem, Israel, June 1990. The MIT Press.

[33] M. Gelfond and H. Przymusinska. Formalization of inheritance reasoning in autoepistemic logic. *Fundamenta Informaticae*, 13:430–443, 1990.

[34] M. Gelfond, H. Przymusinska, and T. Przymusinski. On the relationship between circumscription and negation as failure. *Artificial Intelligence*, 38:75–94, 1989.

[35] M.L. Ginsberg, editor. *Readings in Nonmonotonic Reasoning.* Morgan Kaufmann Publishers, Los Altos, 1987.

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

[37] G. Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2(3):397–425, June 1992.

[38] G. Gottlob. The power of beliefs or translating default logic into standard autoepistemic logic. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 570–575, Chambéry, France, August 1993. Morgan Kaufmann Publishers.

[39] N. Helft, K. Inoue, and D. Poole. Query answering in circumscription. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 426–431, Sydney, Australia, August 1991. Morgan Kaufmann Publishers.

[40] G.E. Hughes and M.J. Cresswell. *A Companion to Modal Logic.* Methuen and Co., London, 1984.

[41] T. Imielinski. Results on translating defaults to circumscription. *Artificial Intelligence*, 32:131–146, 1987.

[42] U. Junker. A correct non-monotonic ATMS. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 1049–1054, Detroit, Michigan, USA, August 1989. Morgan Kaufmann Publishers.

[43] U. Junker and G. Brewka. Handling partially ordered defaults in TMS. In *Proceedings of the European Conference on Symbolic and Quantitative Approaches for Uncertainty*, pages 211–218, Marseille, France, 1991. Lecture Notes in Computer Sciences 548, Springer Verlag.

[44] U. Junker and K. Konolige. Computing the extensions of autoepistemic and default logics with a truth maintenance system. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 278–283, Boston, MA, USA, July 1990. The MIT Press.

[45] A.C. Kakas and P. Mancarella. Generalized stable models: a semantics for abduction. In *Proceedings of the 9th European Conference on Artificial Intelligence*, pages 385–391, Stockholm, Sweden, August 1990. Pitman Publishing.

[46] J. Kalinski. Weak autoepistemic reasoning. Presented at the European Workshop on Theoretical Foundations of Knowledge Representation and Reasoning held in conjunction with ECAI'92, August 3–4, Vienna, Austria, 1992.

[47] M. Kaminski. Embedding a default system into nonmonotonic logic. *Fundamenta Informaticae*, 14:345–354, 1991.

[48] H. Kautz. The logic of persistence. In *Proceedings of the 5th National Conference on Artificial Intelligence*, pages 401–405, Philadelphia, USA, August 1986. Morgan Kaufmann Publishers.

[49] H. Kautz and B. Selman. Hard problems for simple default logics. *Artificial Intelligence*, 49:243–279, 1991.

[50] P. Kolaitis and C. Papadimitriou. Some computational aspects of circumscription. *Journal of the Association for Computing Machinery*, Vol. 37, No. 1, January 1990, pages 1–14, 1990.

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

[52] K. Konolige. On the relation between autoepistemic logic and circumscription. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 1213–1218, Detroit, Michigan, USA, August 1989. Morgan Kaufmann Publishers.

[53] K. Konolige. Quantification in autoepistemic logic. Technical Note 510, SRI International, Menlo Park, California, USA, September 1991.

[54] P. Kuhna. Circumscription and minimal models for propositional logics. In *Proceedings of the First Workshop on Theorem Proving with Analytic Tableaux and Related Methods*, pages 143–155, 1993.

[55] P. Lamarre. A Promenade from Monotonicity to Non-Monotonicity Following a Theorem Prover. In *Principles of Knowledge Representation and Reasoning, Proceedings of the Third International Conference*, pages 572–580, 1992.

[56] P. Lamarre. A Tableaux Prover for Conditional Logics. In *Principles of Knowledge Representation and Reasoning, Proceedings of the Fourth International Conference*, pages ?–580, 1993.

[57] F. Levy. Computing Extensions of Default Theories. In *Proceedings of the European Conference on Symbolic and Quantitative Approaches for Uncertainty*, pages 219–226, Marseille, France, 1991. Lecture Notes in Computer Sciences 548, Springer Verlag.

[58] D. Lewis. Counterfactuals. Basil Blackwell Ltd, 1986, (first edition 1973, Oxford).

[59] V. Lifschitz. Computing circumscription. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 121–127, Los Angeles, California, USA, August 1985. Morgan Kaufmann Publishers.

[60] V. Lifschitz. On the declarative semantics of logic programs with negation. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 177–192. Morgan Kaufmann Publishers, Los Altos, 1988.

[61] V. Lifschitz. Formal theories of action. In M.L. Ginsberg, editor, *Readings in Nonmonotonic Reasoning*, pages 410–432. Morgan Kaufmann Publishers, Los Altos, 1987.

[62] V. Lifschitz. Between circumscription and autoepistemic logic. In *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning*, pages 235–244, Toronto, Canada, May 1989. Morgan Kaufmann Publishers.

[63] V. Lifschitz. Nonmonotonic databases and epistemic queries. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 381–386, Sydney, Australia, August 1991. Morgan Kaufmann Publishers.

[64] F. Lin and Y. Shoham. Epistemic semantics for fixed-point non-monotonic logics. In *Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 111–120, Pacific Grove, USA, March 1990. Morgan Kaufmann Publishers.

[65] F. Lin and Y. Shoham. A logic of knowledge and justified assumptions. *Artificial Intelligence*, 57:271–289, 1992.

[66] W. Marek, G.F. Shvarts, and M. Truszczyński. Modal nonmonotonic logics: Ranges, characterization, computation. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 395–404, Cambridge, MA, USA, April 1991. Morgan Kaufmann Publishers.

[67] W. Marek, G.F. Shvarts, and M. Truszczyński. Modal nonmonotonic logics: Ranges, characterization, computation. *Journal of the ACM*, 40(4):963–990, 1993.

[68] W. Marek and V.S. Subrahmanian. The relationship between logic program semantics and non-monotonic reasoning. In *Proceedings of the 6th International Conference on Logic Programming*, pages 600–617, Lisbon, Portugal, June 1989. The MIT Press.

[69] W. Marek and V.S. Subrahmanian. The relationship between stable, supported, default and autoepistemic semantics for general logic programs. *Theoretical Computer Science*, 103:365–386, 1992.

[70] W. Marek and M. Truszczyński. Relating autoepistemic and default logics. In *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning*, pages 276–288, Toronto, Canada, May 1989.

[71] W. Marek and M. Truszczyński. Modal logic for default reasoning. *Annals of Mathematics and Artificial Intelligence*, 1:275–302, 1990.

[72] W. Marek and M. Truszczyński. Modal logic for default reasoning. In *Preprints of the 3rd International Workshop on Nonmonotonic Reasoning*, pages 247–258, South Lake Tahoe, California, May 1990.

[73] W. Marek and M. Truszczyński. Autoepistemic logic. *Journal of the ACM*, 38:588–619, 1991.

[74] W. Marek and M. Truszczyński. Computing intersection of autoepistemic expansions. In *Proceedings of the 1st International Workshop on Logic Programming and Nonmonotonic Reasoning*, pages 37–50, Washington, D.C., USA, July 1991. The MIT Press.

[75] W. Marek and M. Truszczyński. More on modal aspects of default logic. *Fundamenta Informaticae*, 17:99–116, 1992.

[76] W. Marek and M. Truszczyński. *Nonmonotonic Logic, Context-Dependent Reasoning.* Springer-Verlag, Berlin, 1993.

[77] J. McCarthy. Epistemological problems of artificial intelligence. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pages 1038–1044, Cambridge, MA, USA, August 1977.

[78] J. McCarthy. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

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

[80] J. McCarthy and P.J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, Edinburgh, 1969.

[81] D. McDermott. Non-monotonic logic II. *Journal of the ACM*, 29:33–57, 1982.

[82] D. McDermott and J. Doyle. Non-monotonic logic I. *Artificial Intelligence*, 13:41–72, 1980.

[83] M. Minsky. A framework for representing knowledge. In J. Haugeland, editor, *Mind Design*, pages 95–128. The MIT Press, Cambridge, 1981.

[84] R.C. Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25:75–94, 1985.

[85] R.C. Moore. Autoepistemic logic. In P. Smets, E.H. Mamdani, and D. Dubois, editors, *Non-Standard Logics for Automated Reasoning*, pages 105–136. Academic Press, London, 1988.

[86] I. Niemelä. Decision procedure for autoepistemic logic. In *Proceedings of the 9th International Conference on Automated Deduction*, pages 675–684, Argonne, USA, May 1988. Springer-Verlag.

[87] I. Niemelä. Towards automatic autoepistemic reasoning. In *Proceedings of the European Workshop on Logics in Artificial Intelligence—JELIA '90*, pages 428–443, Amsterdam, The Netherlands, September 1990. Springer-Verlag.

[88] I. Niemelä. Constructive tightly grounded autoepistemic reasoning. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 399–404, Sydney, Australia, August 1991. Morgan Kaufmann Publishers.

[89] I. Niemelä. On the decidability and complexity of autoepistemic reasoning. *Fundamenta Informaticae*, 17(1,2):117–155, 1992.

[90] I. Niemelä. A unifying framework for nonmonotonic reasoning. In *Proceedings of the 10th European Conference on Artificial Intelligence*, pages 334–338, Vienna, Austria, August 1992. John Wiley.

[91] I. Niemelä. A decision method for nonmonotonic reasoning based on autoepistemic reasoning. In *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning*, pages 473–484, Bonn, Germany, May 1994. Morgan Kaufmann Publishers.

[92] I. Niemelä and J. Rintanen. On the impact of stratification on the complexity of nonmonotonic reasoning. In *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*, pages 627–638, Cambridge, MA, USA, October 1992. Morgan Kaufmann Publishers.

[93] N. Olivetti. Tableaux and Sequent Calculus for Minimal Entailment. *Journal of Automated Reasoning*, vol. 9, pages 99–139, 1992.

[94] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36:27–47, 1988.

[95] H. Przymusinska and T.C. Przymusinski. Semantic issues in deductive databases and logic programs. In R. Banerji, editor, *Formal Techniques in Artificial Intelligence*, pages 321–367. North-Holland, Amsterdam, 1990.

[96] T.C. Przymusinski. Three-valued formalizations of non-monotonic reasoning and logic programming. In *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning*, pages 341–348, Toronto, Canada, May 1989. Morgan Kaufmann Publishers.

[97] T.C. Przymusinski. An algorithm to compute circumscription. *Artificial Intelligence*, 38:49–73, 1989.

[98] T.C. Przymusinski. Autoepistemic logics of closed beliefs and logic programming. In *Proceedings of the 1st International Workshop on Logic Programming and Non-monotonic Reasoning*, pages 3–20, Washington, D.C., USA, July 1991. The MIT Press.

[99] T.C. Przymusinski. Three-valued nonmonotonic formalisms and semantics of logic programming. *Artificial Intelligence*, 49:309–343, 1991.

[100] M. Reinfrank, O. Dressler, and G. Brewka. On the relation between truth maintenance and autoepistemic logic. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 1206–1212, Detroit, Michigan, August 1989. Morgan Kaufmann Publishers.

[101] R. Reiter. On closed world data bases. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 55–76. Plenum Press, New York, 1978.

[102] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[103] R. Reiter. Nonmonotonic reasoning. *Ann. Rev. Comput. Sci.*, 2:147–186, 1987.

[104] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

[105] R. Reiter. On asking what a database knows. In *Proceedings of the Symposium on Computational Logic*, pages 95–113, Brussels, Belgium, November 1990. Springer-Verlag.

[106] J.S. Schlipf. How uncomputable is general circumscription? In *Proceedings of the Symposium on Logic in Computer Science*, pages 92–95, Cambridge, USA, June 1986. IEEE Computer Society Press.

[107] G. Schwarz and M. Truszczyński. Nonmonotonic reasoning is sometimes simpler. In *Proceedings of the 3rd Kurt Gödel Colloquium on Computational Logic and Proof Theory*, pages 313–324, Brno, Czech Republic, August 1993. Springer-Verlag.

[108] G. Shvarts. Autoepistemic modal logics. In *Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 97–109, Pacific Grove, USA, March 1990. Morgan Kaufmann Publishers.

[109] C. B. Schwind. A tableau-based theorem prover for a decidable subset of default logic. In M. E. Stickel, editor, *Proceedings of the 10th International Conference on Automated Deduction*, number 449 in Lecture Notes in Artificial Intelligence, pages 541–546. Springer Verlag, July 1990.

[110] C. B. Schwind and V. Risch. A tableau-based characterisation for default logic. In R. Kruse and P. Siegel, editors, *Proceedings of the European Conference on Symbolic and Quantitative Approaches for Uncertainty*, number 548 in Lecture Notes in Computer Science, pages 310–317. Springer Verlag, October 1991.

[111] C. B. Schwind and V. Risch. Tableau-based characterisation and theorem proving for default logic. *Journal of Automated Reasoning*, 1994. to appear.

[112] C. B. Schwind and P. Siegel. Modal logic for hypothesis theory. *Fundamenta Informaticae*, vol. 20, No. 1, 1994.

[113] P. Siegel and C. B. Schwind. A modal based theory for nonmonotonic reasoning. *Journal of Applied Non-Classical Logics*, vol. 3, No. 1, pages 73–92, 1993.

[114] R. M. Smullyan. *First Order Logic*. Springer, 1968.

[115] R.F. Stärk. On the existence of fixpoints in Moore's autoepistemic logic and the non-monotonic logic of McDermott and Doyle. In *Proceedings of the 4th Workshop on Computer Science Logic*, pages 354–365, Heidelberg, Germany, October 1990. Springer-Verlag.

[116] J. Stillman. It's not my default: the complexity of membership problems in restricted propositional default logics. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 571–578, Boston, Massachusetts, USA, July 1990. The MIT Press.

[117] J. Stillman. The complexity of propositional default logics. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 794–800, San Jose, California, USA, July 1992. The MIT Press.

[118] M. Truszczyński. Embedding default logic into modal nonmonotonic logics. In *Proceedings of the 1st International Workshop on Logic Programming and Nonmonotonic Reasoning*, pages 151–165, Washington, D.C., USA, July 1991. The MIT Press.

[119] M. Truszczyński. Modal interpretations of default logic. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 393–398, Sydney, Australia, August 1991. Morgan Kaufmann Publishers.