Nonmonotonic Reasoning

V.W. Marek and A. Nerode

Classical logic is the study of "safe" formal reasoning. Western Philosophers developed classical logic over a period of thirty-three centuries after its introduction in the form of syllogistic by Aristotle [1] in the third century B. C. Beginning in the nineteenth century with De Morgan [2] and Boole [3], responsibility for the development of classical logic moved from the philosophical to the mathematical community. Boole [3], Peirce [4], Frege [5], Schröder [6], Russell [7], Hilbert [8], Gödel [9], and Tarski [10] showed how to make classical logic into a mathematical discipline which led to model theory, set theory, recursion theory, and proof theory. The discovery of recursive function theory by Church [11], Gödel [12], Herbrand [13], Kleene [14], and Turing [15] in the nineteen thirties, the development of digital computers with a Von Neumann architecture in the 1950's, the interest in machine simulation of human intelligence (AI) in the 1960's, and the development of mathematical linguistics led to many new applications of classical logic. The logical needs of these subjects outstrip all previously existing developments and present many new challenges which require non-traditional logics tailored to computer science. New problems suggested by computer science and artificial intelligence have led to an unprecedented multiplicity of intellectual challenges. As a result there now more research effort in logics for computer science than there ever was in traditional logics.

Some of the needed logics are extensions of classical logic by new operators. Example are modal and temporal logic [16, 17]. These are useful as logics of programs and logics of agents. Some needed logics are restrictions of classical logic. Examples are intuitionistic logic which can be used to extract programs from proofs and ideal PROLOG [18], where the program specification is the program itself. Some needed logics develop aspects of higher-order logics. These received little attention previously. These include Huet-Coquand's Theory of Constructions [19], based on Girard's strong normalization theorem for higher order intuitionistic logics [20]. Another is Constable's NUPRL [21], based on the normalization theorem of Martin-Löf's predicative higher order logics [22]. All thus far mentioned are extensions or restrictions of classical logics.

An outstanding feature of classical logics and their extensions and restrictions is that once the premises of an argument have been accepted and an inference of a conclusion from these premises has been made, that conclusion enters the body of knowledge and is never withdrawn so long as the premises are maintained. This gives rise to a unique deductive closure of the set of premises, consisting of all deductive consequences of the premises. Thus it was that we have accumulated over thousands of years a larger and larger body of theorems in classical mathematics, all consequences of a few premises which are now called the axioms of set theory by a fixed set of rules of inference, all part of the deductive closure of the premises using the logical calculi taught in the conventional courses of logic. These logics are "monotone", that is, conclusions, once established, are never retracted. Larger sets of premises give larger sets of conclusions.

Non-monotone logics have been developed recently which describe commonsense reasoning which is neither a restriction nor an extension of classical logic. Consequences of premises are drawn as much due to the absence as to the presence of knowledge. When more knowledge is acquired, conclusions previously drawn may have to be withdrawn because the rules of inference that led to them no longer are active. Intelligent decision makers use this form of commonsense reasoning to infer actions to be performed from premises which cannot be made by classical logic inference, because they simply have to make decisions whether or not there is enough information for a classical logical deduction.

To see what this means, we have to explain in what sense classical logic and its restrictions and extensions encompass all "safe" modes of reasoning. A "safe" mode of reasoning is one in which every conclusion drawn from premises by this mode of reasoning is true in all intended interpretations (or models) in which the premises are true. A ”completeness and correctness theorem” for a system says that the ”safe” rules of deduction in the textbooks generate exactly all those conclusions from premises which are true in every interpretation in which all the premises are true. Thus any extension or restriction of classical logic with a notion of interpretation or model and a completeness and correctness theorem for that notion does not need any new ”safe” rules of inference, except perhaps to expedite deductions.

So the extra element in commonsense reasoning that allows different conclusions to be drawn than in classical logics is the use of rules of inference which are not ”safe”. They are to be used not because they are safe or unsafe, but because they usually give conclusions useful for decision making that can not otherwise be obtained. For a rule to be ”unsafe” means there exist interpretations, or states of the world, in which the premises of the rule hold, but the conclusions do not. The intention is that we use such rules when we expect exceptions to be rare. Such unsafe rules of inference are characteristically used when conclusions must be drawn and decisions made but our knowledge and past experience is too limited, too uncontrolled, or too unmodelled to draw a decision by classical logical or statistical inference. Formalized commonsense reasoning is provide a a principled method of ”jumping to conclusions” based on premises that are merely ”rules of thumb”.

Formalization of commonsense reasoning can be said to have started with Aristotle’s introduction of the modal logic of possibility and necessity. C. I. Lewis formalized such logics in the Monist in 1915 and in his Survey of Symbolic Logic with Langford in 1928. More recently, there has been much activity in formalizing the logics of "I know ...", and "I believe... P". But it is only since 1980 or so, under the influence of John McCarthy, that non-monotonic reasoning as such has been systematically formalized. McCarthy, building on the efforts of earlier philosophers stemming from those mentioned above, showed how to formalize at least some commonsense reasoning [23]. This discovery formed the impetus for pioneering investigations of of Reiter, [24, 25], McDermott and Doyle, [26, 27] and others. All this led to the development of what is now a popular area of research, *Nonmonotonic Logics* – which we shall outline in this article.

We believe that the development and computer implementation of non-monotonic systems is a necessary prolegomena to the development of future intelligent systems capable of simulating higher human cognitive functions.

1 Classical Logic

Classical Logic, be it in its propositional fragment, predicate fragment or other logics (modal and intuitionistic ones) is based on the notion of consequence. In this general framework a logic is represented by its *syntax* i.e. the set of its well-formed formulas and *semantics* which provides the meaning to that syntax. Thus, usually, we have a certain set of formulas constructed inductively from some primitive (atomic) formulas by means of appropriate functors. Next we assign to such language a *semantics* These are valuations of propositional variables (in case of propositional logic), relational structures (in case of predicate logic) or Kripke structures (for intuitionistic and modal logics). Semantics always generates a *semantic consequence* relation defined by means of *semantic entailment*. Let us see how it works in the case of propositional logic. We say that a formula φ is a semantic consequence of a set of formulas T (is semantically entailed by, in symbols $T \models \varphi$) if every valuation satisfying every formula from T, satisfies φ as well. The propositional logic is decidable, that is there is an algorithm for testing if a formula φ is entailed by a finite set of formulas, T. The technique of *tableaux* provides one such method. Moreover we can list all the valuations of variables appearing in T and φ and check them. In the case of predicate logic such technique is not available. The reason is that one has to take into account infinite relational structure. Moreover there is infinitely many of them. This implies that the semantic entailment relation $T \models \varphi$ is not effective. Therefore it is desirable to have a *syntactic* technique for testing entailment. Specifically we seek methods that use formula manipulation to test entailment. This is done by means of *provability*. This relationship is denoted by $T \vdash \varphi$. There are various techniques for the syntactic entailment. These include: natural deduction, [28], Hilbert-style systems, [8], resolution refutation, [29] and semantic tableaux, [30, 31]. It always involves manipulation of formulas and derivation of additional formulas by syntactic means. The key result in such techniques is always a *completeness* property which says that $T \models \varphi$ if and only if $T \vdash \varphi$. There are several completeness theorems, for each of the logics mentioned above separately. Since the proofs are finite, completeness property implies *compactness* property, that is the statement that $T \models \varphi$ if and only if for some *finite* subtheory $T' \subseteq T$, $T' \models \varphi$. This also implies that the set of consequences of a theory T is recursively enumerable in T.

Let us look now at the abstract form of the consequence operation described above. Let $Cn(T)$ be the set of semantic consequences of the theory T. Since a larger theory has less models, and T is the intersection of the sets of formulas true in all these models, we find the following property of theories, called *monotonicity*

$$T_1 \subseteq T_2 \text{ implies } Cn(T_1) \subseteq T_2$$

Consequence operations have other properties as well, for instance it is easy to see that Cn is *idempotent*, that is $Cn(Cn(T) = Cn(T)$, but monotonicity is a fundamental property. It tells us that once we established a fact on the basis of our theory, we will never have to withdraw it as long as we only add new premises. Any future additional observations can only confirm it. This property governs the way our knowledge accumulates. Whatever is *proved* using logic remains true. Thus, in spite of the fact that our knowledge grows, the formal results in Mathematics accumulate. Whatever was proved, stays proved forever as long as the assumptions grow.

2 Commonsense Reasoning

Although the logical consequences of formulas in which we believe should also be believed (after all, if we believe that all men are mortal and Socrates is a man, then we have to believe that Socrates is mortal), in commonsense we often employ, in addition to classical logic reasonings, some other methods of reaching conclusions. There are numerous types of argumentation used in commonsense reasoning. For instance we often make a tacit assumptions that we have a complete information about some fact. Then, it is enough to list explicitly only those which are true – the remaining are inferred false by our tacit assumption. Clearly such reasoning is not monotonic. Indeed, let T consists of facts that are assumed to be true. Now, if we add another fact, not in T, then previously it was assumed to be false, but now it is true, so the monotonicity property is violated. Similarly, we often reason using "default reasoning", that is we make assertions simply because we do not have information which blocks making such inferences. But, again, such reasoning cannot be monotonic. If our beliefs are augmented by new facts blocking the inferences, then the inference which led to the specific conclusion may be blocked. Yet another method of reasoning (related to the the previous one) is reasoning from both belief and absence of thereof. If we use in our reasoning the fact that we do not have a fact among beliefs then our reasoning may become invalid if a new belief is asserted. Yet another method of reasoning is when we reason about the effects of actions. We then tend to think that only those aspects of the world that are directly related to the action performed could have changed. But, of course, such reasoning must be nonmonotonic, as additional things could change by unrelated reasons.

The formal proposals which we will discuss in the subsequent sections of this review address the technical developments addressing the above modes of reasoning. What is most amazing in the theory we present is that they can be formalized at all. Thus we will present Closed World Assumption, Default Logic, Modal Nonmonotonic Logics, and Circumscription. We will also discuss some nonmonotonic aspects of Logic Programming and a general mechanism for treating nonmonotonicity in logic.

The subject of this article is discussed in several monographs. These include: [32, 33, 34, 35, 36, 31]. The theory of nonmonotonic reasoning is, at present, an active area of research in Artificial Intelligence and many monographs will, undoubtedly, follow.

3 Closed World Assumption

Closed World Assumption (*CWA* for short) is historically the earliest form of nonmonotonic reasoning. It is due to Reiter, [24]. Reiter analyzed the way in which information is extracted out of databases and realized that a database contains, implicitly, a great wealth of *negative* information. That is, every elementary fact that can be stored in a database but is not, is assumed to be false. This is the reason why, when asking an airline phone operator for a flight from New York City to San Francisco arriving at 7:45 am you get either yes or no answer, but (in principle) no "I do not know" for an answer. The reason is that the lack of information is processed as falsity. More formally, given a database encoded as a first-order theory T, define, for an atomic ground statement p

$$T \vdash_{CWA} \neg p \ \text{ if } \ T \nvdash p$$

Then define $CWA(T) = Cn(T \cup \{\neg p : \ p \ \text{ is a ground atom, } \ T \vdash_{CWA} \neg p\})$.

When T is a propositional theory then $CWA(T)$ is a complete theory. The reason is that for each atom p, $p \in CWA(T)$ or $\neg p \in CWA(T)$.

In general, CWA is not a safe mode of reasoning. In fact for a consistent theory T $CWA(T)$ may be inconsistent. For instance the theory $T = \{p \vee q\}$ is consistent whereas $CWA(T)$ is inconsistent. Generally, CWA handles disjunctive information poorly. The basic result on reasoning under CWA, due to Reiter, [24], is that for a Horn theories (that is theories consisting of clauses which have at most one positive literal) CWA is a safe mode of reasoning. That is, if T is consistent and Horn theory, then $CWA(T)$ is consistent too.

The operator described by CWA is nonmonotonic. That is $T_1 \subseteq T_2$ does not imply $CWA(T_1) \subseteq CWA(T_2)$. This follows immediately from our remark that $CWA(T)$ is complete in propositional case.

CWA is closely related to Logic Programming. Specifically, if P is a (Horn) logic program, P_{ground} is the set of ground instances of clauses of P, then $CWA(P_{ground})$ is precisely the propositional theory of the least Herbrand model of P. This implies that the answers to the ground atomic queries to P coincide with the results of SLD resolution with respect to P. More information on CWA and Logic Programming can be found in, [37].

4 Default Logic

Default logic is one of the better understood formalisms in Nonmonotonic Reasoning. It admits various interpretations, but the most popular is that it assigns meaning to the quantifier "under usual circumstances". The idea here is to capture the meaning of this concept not by means of a probabilistic interpretation, but rather by controlling how the derivations are made, that is by syntactic means. The derivations are made by usual type reasoning systems, except that the rules carry the list of "exceptional cases" making the application of such rule invalid.

Formally, Reiter, [25] introduced the concept of default theory. A default theory is a pair $\langle D, W\rangle$ where W is a set of sentences of the underlying language $\mathcal{L}$ and D consists of *default rules*. A default rules is an entity of the form

$$r = \frac{\alpha : M\beta_1, \ldots, M\beta_k}{\gamma} \tag{1}$$

where $\alpha, \beta_1, \ldots, \beta_k, \gamma$ are sentences of the underlying language $\mathcal{L}$.

A rule of the form (1) is sometimes interpreted as: "if α has been established, and all $\beta_1, \ldots, \beta_k$ are all possible then derive γ". The main issue here is how we interpret the word "possible". Specifically, possible with respect to what? Since the first-order logic (be it propositional or predicate) offer a way of interpreting "possible β" as "possible with respect to a theory T" (namely as $T \not\vdash \neg\beta$) the question arises which T should be selected. Reiter offered an elegant solution to this problem. In order to formally define the way the rules are applied and, subsequently, define the consequences of default theory we introduce a notion of T-consequences of a default theory $\langle D, W\rangle$. Namely, if $T \subseteq \mathcal{L}$, then the T-consequences of $\langle D, W\rangle$ is the smallest set S of formulas satisfying these conditions:

1. $W \subseteq S$
2. $Cn(S) = S$ (here Cn is the consequence relation of $\mathcal{L}$)
3. For all rules $r \in D$ of the form (1), if $\alpha \in S$ and $\neg\beta_1 \notin Cn(T), \ldots, \neg\beta_k \notin Cn(T)$ then $\gamma \in S$.

Clearly, such least set exists for any T. T controls which rules are applicable in the process of making the derivation. Following Reiter denote $\Gamma(T)$ the set so constructed. Clearly, Γ depends on both D and W.

Now we can see why we were talking about reasoning "in usual circumstances". Namely, the rule of the form (1) carries within itself the description of the exceptions of its applicability. These exceptions are: $\neg\beta_1, \ldots, \neg\beta_k$. None of the exception may happen if we want the rule applied. The context, T tells us where are we supposed to look for information concerning the applicability. Thus T is mentally perceived as the state of affairs. The derivations can be made in its presence. Now we have to define which are “correct” states of affairs from the point of view of a default theory $\langle D, W \rangle$.

Reiter, [25] defines an extension as the fixpoint of the operator Γ that is any solution of the equation $\Gamma(T) = T$. Notice that $\Gamma(T) = \Gamma(D, W, T)$, that is both D, W contribute to the definition. The operator Γ is monotonic in D and in W but it is antimonotonic in T. That is, with D, W fixed,

$$T_1 \subseteq T_2 \text{ implies } \Gamma(T_2) \subseteq \Gamma(T_1)$$

Thus the usual ways of computing a fixpoint fail and there is no guarantee that a fixpoint exist.

Notice that our analysis of the operator Γ implies that the equation $\Gamma(T) = T$ means two things:

(a) Every formula in $\Gamma(T)$ belong to T. That is all formulas which have a derivation from $\langle D, W \rangle$ using T as a controlling context belong to T. This means that nothing outside of T will be derived.

(b) Every formula in T possesses a derivation with T serving only as a controlling context for applicability of rules. That is all the formulas in T can be reconstructed from W using underlying logic and those rules of D which are not blocked by T.

As noticed above, there is no guarantee that a default theory possesses an extension. Also, if extensions exist they may be multiple. Hence, in opposition to classical logic, default logic assigns to a default theory a single, multiple or no consequences at all.

Given a default rule of the form (1), define $c(r) = \gamma$ and $c(D) = \{c(r) : r \in D\}$. Then it can easily be proved that for every S, $\Gamma(S) \subseteq Cn(W \cup c(D))$. This is a "bounding principle" which allows us to compute examples.

Example 1 *1.* $W = \{p\}, D = \{\frac{p:M\neg q}{q}\}$ *where* p, q *are distinct propositional variables. Then according to the above bounding principle there are only two candidates for extensions of* $\langle D, W\rangle$, $T_1 = Cn(\{p\})$ *and* $T_2 = Cn(\{p, q\})$. *Since* $\neg\neg q \notin T_1$, $\Gamma(T_1) = Cn(\{p, q\}) \neq T_1$. *Thus* T_1 *is not an extension of* $\langle D, W\rangle$. *On the other hand* $\neg\neg q \in T_2$, $\Gamma(T_2) = Cn(\{p\}) \neq T_2$. *Thus* T_2 *is also not an extension of* $\langle D, W\rangle$ *and so* $\langle D, W\rangle$ *has no extension.*

2. $W = \{p\}, D = \{\frac{p:M\neg q}{r}\}$ *where* p, q *and* r *are distinct propositional variables. Again we have two candidates for an extension of* $\langle D, W\rangle$, $T_1 = Cn(\{p\})$ *and* $T_2 = Cn(\{p, r\})$. *It is easy to see that* $\Gamma(T_1) = Cn(\{p.r\}) \neq T_1$. *But again,* $\Gamma(T_2) = Cn(\{p, r\}) = T_2$. *Thus* T_2 *is a unique extension of* $\langle D, W\rangle$.

3. $W = \{p\}, D = \{\frac{p:M\neg q}{r}, \frac{p:M\neg r}{q}\}$ *where* p, q *and* r *are distinct propositional variables. The reader will check easily that there are four candidates for extensions out of which two,* $T' = Cn(\{p, q\})$ *and* $T'' = Cn(\{p, r\})$ *are extensions of* $\langle D, W\rangle$.

If S is a set of formulas then a default rule of the form (1) is called a generating default for S if $\alpha \in S$, $\neg\beta_1 \notin S, \ldots, \neg\beta_k \notin S$. $GD(D,S)$ is the set of all generating defaults for S belonging to D.

We will formulate a number of fundamental properties of default extensions.

1. (Antichain property) If S_1, S_2 are distinct extensions of $\langle D, W \rangle$ then $S_1 \not\subseteq S_2$.

2. (Confirmation of evidence) If S is an extension of $\langle D, W \rangle$ and $W' \subseteq S$ then S is an extension of $\langle D, W \cup W' \rangle$.

3. If T is an extension of $\langle D, W \rangle$ then T is a unique extension of $\langle GD(D,T), W \rangle$.

4. If all defaults in D have at least one justification, then $\langle D, W \rangle$ has an inconsistent extension if and only if W is inconsistent.

An important class of default theories consist of *normal default theories.* A rule r of the form (1) is normal if $k = 1$ and $\gamma = \beta_1$. That is normal rules are of the form

$$r = \frac{\alpha : M\gamma}{\gamma} \tag{2}$$

A default theory $\langle D, W \rangle$ is called *normal* if all rules in D are normal. Normal theories have much stronger properties.

1. (Existence) A normal default theory always possesses an extension.

2. (Incompatibility) If S_1, S_2 are two distinct extensions of a normal default theory $\langle D, W \rangle$ then $S_1 \cup S_2$ is inconsistent.

3. (Semi-monotonicity) If $\langle D_1, W\rangle$, $\langle D_2, W\rangle$ are two normal default theories and $D_1 \subseteq D_2$ then for every extension S_1 of $\langle D_1, W\rangle$ there exists an extension $\langle D_2, W\rangle$ such that $S_1 \subseteq S_2$.

4. If $\langle D, W\rangle$ is a normal default theory and $W \cup c(D)$ is consistent then $\langle D, W\rangle$ possesses a unique extension.

Closed World Assumption (discussed in Section 3) can be represented in normal default logic by the following embedding. Assign to a theory S a default theory $\langle D_{CWA}, S\rangle$ where D_{CWA} consists of all defaults $\frac{:M\neg p}{\neg p}$ for p being an atom of the language $\mathcal{L}$. Then S is CWA-consistent if and only if $\langle D_{CWA}, S\rangle$ possesses a unique extension. Moreover, in that case $CWA(S)$ is the unique extension of $\langle D_{CWA}, S\rangle$.

Default logic generalizes classical logic. If $D = \emptyset$ then a unique extension of $\langle D, W\rangle$ is $Cn(W)$. It turns out that default logic shares other properties with classical logic. For instance default logic possesses several normal forms, [38].

Besides normal defaults one often considers seminormal defaults. These rules are of the form

$$r = \frac{\alpha : M(\beta \wedge \gamma)}{\gamma} \tag{3}$$

Default theories such that every rule has no justification or i seminormal represent default logic faithfully. Specifically, for every default theory $\langle D, W\rangle$ there is a default theory $\langle D', W\rangle$ such that every rule in D' is seminormal or has no justification and with the following properties. First, every extension of $\langle D, W\rangle$ is included in some extension of $\langle D', W\rangle$. Second there is one-to-one correspondence between extensions of $\langle D', W\rangle$ and extensions of $\langle D, W\rangle$.

There are many algorithms for computing extensions of default theories. All of these are inefficient because the complexity problems associated with default logic are located, generally, on the second level of the polynomial hierarchy. Specifically, Gottlob, [39] found the complexity of basic problems for default logic

1. (Existence) The problem: Given a default theory $\langle D, W\rangle$ does $\langle D, W\rangle$ possesses an extension, is Σ_2^P-complete.

2. (Membership in some extension) The problem: Given a default theory $\langle D, W\rangle$ and a propositional formula φ, does there exist an extension S of $\langle D, W\rangle$ such that $\varphi \in S$ is Σ_2^P complete.

3. (Membership in all extensions) The problem: Given a default theory $\langle D, W\rangle$ and a propositional formula φ, does φ belong to all extension of $\langle D, W\rangle$ is Π_2^P complete.

In the case of normal default theories the first problem is simple, because every normal default theory possesses an extension. The second and the third problem have the same complexity.

Default logic is a formalism based, primarily on extension of syntactic constructions of first order logic. Nevertheless there are several semantical characterizations of extensions. Lifschitz, [40] and Guerreiro and Casanova, [41] introduced semantical constructions characterizing expansions.

There has been numerous proposals for modifications of default logic. None of these is widely accepted. Of various proposals, we mention Łukaszewicz, [42] and Brewka, [34]. These proposals suggested changing the mechanism for computing objects associated with default theories. Other structures associated with default theories include well-founded extensions of Baral and Subrahmanian, [43], weak extensions Marek and Truszczynski, [44] and many others.

Default logic is, currently, best understood among several modes of nonmonotonic reasoning. There are several monographs devoted to the subject. These include: Besnard, [32], Łukaszewicz, [36], Brewka, [45] and Marek and Truszczynski, [35].

5 Nonmonotonic Aspects of Logic Programming

A Horn logic program is a finite set of expressions of the form

$$p(\bar{X}) \leftarrow q_1(\bar{(}X)), \ldots, q_m(\bar{X}) \tag{4}$$

Here $\bar{X}$ is a string of variables. Not all variables must appear in all predicates. An expression (4) is called a *clause*, $p(\bar{X})$ is the head of that clause whereas $q_1(\bar{(}X)), \ldots, q_m(\bar{X})$ is the body of the clause. Such clause possesses a *logical interpretation*. It is a formula

$$\forall_{\bar{X}}(q_1(\bar{(}X)) \wedge \ldots \wedge q_m(\bar{X}) \supset p(\bar{X}))$$

ground(P), where P is a program, is the family of all ground substitutions of clauses from P.

With a program P we associate an operator T_P, mapping subsets of Herbrand base into subsets of Herbrand base, by a the following

$$T_P(I) = \{p : \exists_{C \in ground_P} I \models body(C) \wedge p = head(C)\}$$

van Emden and Kowalski, [46]. Operator T_P is monotone and compact. Hence, by Knaster-Tarski Lemma, T_P possesses the least fixpoint. This fixpoint coincides with the least Herbrand model M_P of P. Moreover, if P_1, P_2 are two programs, $P_1 \subseteq P_2$ then $M_{P_1} \subseteq M_{P_2}$.

When we introduce negation into the body of clauses of the program, we get *general logic programs*. They consist of clauses of the form:

$$p(\bar{X}) \leftarrow q_1(\bar{(}X)), \ldots, q_m(\bar{X}), \neg r_1(\bar{X}), \ldots, \neg r_n(\bar{X}) \qquad (5)$$

The notion of the logical interpretation and of the operator T_P generalize directly to general logic program in a natural fashion.

Although a general logic program always possesses a model (Herbrand base is a model of every program), the existence of the least Herbrand model is no longer guaranteed. There are, however, always minimal models of general logic program. There may be several minimal models of a general programs. The question how to assign the meaning of a general program is one of the most important issues in foundations of logic programming. We will look at several proposals for assigning such meaning.

5.1 Minimal models

As noticed above minimal models of a program always exist. However, a minimal model may be fairly artificial and may not be connected with the process of computation. To see that look at the program P_1 consisting of a single clause $p \leftarrow \neg q$. Intuitively the "correct" model of P is $\{p\}$ (q cannot be computed, so p can be), but P possesses two minimal models (the other is $\{q\}$). Moreover, adding new clauses to a program changes semantics drastically. A minimal model of a larger program does not need to be minimal as a model of a smaller program. For complexity results on minimal models see Eiter and Gottlob, [47].

5.2 Supported models

One can assign to a general program its *completion* Clark, [48]. Completion of the program is a first order theory obtained from P by the following procedure. For each predicate p a formula

$$\forall_{\bar{X}} p(\bar{(}X)) \equiv (B_1 \vee \ldots \vee B_k)$$

is the *completion* of p. Here formulas B_j, $1 \leq j \leq k$ are obtained by elimination of terms in the heads of clauses with head p. Completion of a program is the theory consisting of completions of predicates of the programs incremented by axioms about equality (so called Clark Equational Theory) A supported structure for P is a model of completion of P. Such structure is a model of P and so it is called supported model of P. Apt and van Emden, [49] proved that Herbrand models of completion of P are characterized as fixpoints of the operator T_P.

5.3 Perfect model

Some programs allow for identification of a particular minimal model which has particularly nice properties. Call a program P *stratified* Apt, Blair and Walker, [50] if there is a function *rank* on the set of predicates of the program P such that whenever we have a clause of the form (5) then for all i $rank(q_i) \leq rank(p)$ and for all j, $rank(r_j) < rank(p)$. The intuition is that the negative information necessary to compute the extension of the predicate p in the desired model of P must be computed *before* the stage in which p is computed. Checking if P is stratified can be done in time linear in $|P|$. The desired model is computed in stages. First, compute extensions of predicates of rank 0. This information is then used to compute extensions of predicates of rank 1, etc. The resulting model is called the prefect model of P. Not every program is stratified, and so a perfect model is not always defined. If it exists then it is unique and does not depend on a particular stratification used in its construction. Moreover the perfect model of P is a model of completion of P as well. If P is a propositional program that is stratified than its perfect model can be computed in time linear in $|P|$. In predicate case Apt and Blair, [51] show that a program with n strata can compute a Σ^0_n complete set. Conversely the perfect model of a stratified program with n strata is Σ^0_n.

A "local" version of stratification is due to Przymusinski, [52]. Here we require that the rank is defined not on the set of all predicates of the language of the program but rather on the Herbrand base. The stratification conditions are similar, but pertain to clauses in *ground*(P). Again, one can assign to a locally stratified program the perfect model. There is no difference between stratification and local stratification in the case of finite propositional programs. In the predicate case testing if P is locally stratified is Π^1_1-complete, Cholak and Blair, [53], Perfect model of a locally stratified, recursive program is hyperarithmetical and all hyperarithmetical sets are extensions of predicates in perfect model of suitably chosen locally stratified program.

5.4 Stable models

Gelfond and Lifschitz, [54] defined the notion of a stable model of a program. Let M be a subset of the Herbrand base of P. Reduce the program *ground*(P) as follows. Given a clause C

$$p \leftarrow q_1, \ldots .q_m, \neg r_1, \ldots, \neg r_n$$

in *ground*(P) eliminate C altogether if for some j, $r_j \in M$. Otherwise let C^M be

$$p \leftarrow q_1, \ldots .q_m$$

P^M consists of C^M for those C which are not eliminated. Since P^M is a Horn program it possesses a least model N_M. We call M a stable structure for P if $N_M = M$. A stable structure for P (if exists) is a model of P. Moreover it is a model of completion of P and a minimal model of P. If P is stratified or locally stratified then P possesses a unique stable model. It is its perfect model. It turns out that stable models of general logic programs are closely connected with Default Logic, Bidoit and Froidevaux, [55], Marek and Truszczynski, [56]. There exist programs without stable models. The existence problem for stable models of propositional general logic programs is NP-complete, Marek and Truszczynski, [57]. For finite predicate general logic programs existence of a stable model is a Σ^1_1-complete problem Marek, Nerode and Remmel, [58].

6 Modal nonmonotonic logics

The modal flavor of nonmonotonic reasoning has been noticed at the very beginning of developments. In facts, Reiter's notation suggests that he considered the formulas in justification as modalized formulas, at least conceptually. McDermott and Doyle, [26] suggested a construction in the language of modal logic which handles nonmonotonicity well. McDermott and Doyle construction is based on the analysis of the notion of introspection. Let $\mathcal{L}_L$ be a modal language with modal unary functor L. Given a theory T, we say that T is closed under *positive introspection* if

$$\varphi \in T \text{ implies } L\varphi \in T$$

Thus closure under positive introspection is nothing more that closure under necessitation rule of modal logic. Similarly, we say that T is closed under *negative* *introspection* if

$$\varphi \notin T \text{ implies } \neg L\varphi \in T$$

Negative introspection has a distinctly nonmonotonic flavor – if φ is not in T the $\neg L\varphi$ is in T. Theories closed under propositional provability (in the language of modal logic) and both forms of introspection are called *stable*, Stalnaker, [59]. They are precisely theories of Kripke models with universal accessibility relation.

McDermott and Doyle, [26, 27] considered the following operator

$$MDD(T) = Cn_{\mathcal{S}}(I \cup \{\neg L\varphi : \varphi \notin T\})$$

There are two parameters in this operator. First, I – the set of initial beliefs (or initial knowledge) of a reasoning agent. Second, a modal logic $\mathcal{S}$. The fixpoints of that operator, that is solutions of the equation

$$T = Cn_{\mathcal{S}}(I \cup \{\neg L\varphi : \varphi \notin T\})$$

are called $\mathcal{S}$-expansions of I. The consequence operator $Cn_{\mathcal{S}}$ used here is stronger than in usual modal logics – necessitation is applicable to all formulas and not only to axioms of $\mathcal{S}$. The strength of the McDermott and Doyle construction lies in additional parameter $\mathcal{S}$. Varying $\mathcal{S}$ can lead to very different solutions. McDermott noted that as long as $\mathcal{S}$ contains necessitation, every expansion is a stable theory. Moreover an $\mathcal{S}$-expansion is closed under $S5$-consequence. McDermott, [27] noticed that for $\mathcal{S} = S5$, $\mathcal{S}$-expansions of I coincide with theories of $S5$-models of I with universal relations. This implies that the formulas true in all $S5$-expansions of I are precisely $S5$-consequences of I. This was considered counterintuitive at a time.

Moore, [60] considered a different operator, explicitly taking into account both positive and negative introspection with respect to T

$$Mo(T) = Cn(I \cup \{L\varphi : \varphi \in T\} \cup \{\neg L\varphi : \varphi \notin T\})$$

and argued that this operator is more natural than McDermott and Doyle operator. Fixpoints of Moore operator, i.e. solutions to the equation

$$T = Cn(I \cup \{L\varphi : \varphi \in T\} \cup \{\neg L\varphi : \varphi \notin T\})$$

are called autoepistemic (or stable) expansions of I. It turned out, however, that as long as consistent theories are concerned, Moore construction is a special case of McDermott and Doyle construction, namely $\mathcal{S} = KD45$, Schwarz, [61]. Another special case is so-called *reflexive autoepistemic logic* of Schwarz, [62], where reflexive autoepistemic expansions are solutions of the equation

$$T = Cn(I \cup \{L\varphi \equiv \varphi : \varphi \in T\} \cup \{\neg L\varphi : \varphi \notin T\})$$

This is again a special case of McDermott and Doyle construction, corresponding to modal logic $S4.4$ (see Chellas, [16] for more information on modal logics). All the operators mentioned above are nonmonotonic. Logic based on fixpoints of Moore operator is called autoepistemic logic. It is currently believed that this logic properly models beliefs of a fully introspective, reasoning, agent.

The notion of $\mathcal{S}$-expansion behaves monotonically in $\mathcal{S}$. That is, given $I \subseteq \mathcal{L}_L$, $\mathcal{S}_1$ and $\mathcal{S}_2$, if $\mathcal{S}_1 \subseteq \mathcal{S}_2$ then every $\mathcal{S}_1$-expansion of I is also an $\mathcal{S}_2$-expansion of I.

It turns out that different modal logics may generate uniformly in I the same expansions. This happens, for instance for modal logics 5 (subnormal modal logic without the scheme K), $K5$, $K45$, and $KD45$. Many other areas (called ranges) where expansions coincide are known presently. One such area is between $S4$ and $S4F$ for all *finite* theories I Marek, Schwarz and Truszczynski, [63].

The degree of freedom (the choice of $\mathcal{S}$) present in modal nonmonotonic logics makes them a very powerful mechanism for representation of other forms of nonmonotonic reasoning. There are many representations of nonmonotonic reasonings in modal nonmonotonic logics with various degree of faithfulness. These include Konolige interpretation of default logic in autoepistemic logic, [64], Truszczynski interpretation of default logic in nonmonotonic $S4F$, [65] and many others. Stable semantics of logic programs can be faithfully represented in both autoepistemic logic, [66] and in reflexive autoepistemic logic, [67]. Similarly, supported semantics for logic programs can be faithfully represented in autoepistemic logic, [68].

Lifschitz, [69] introduced a general mechanism for uniform treatment of modal nonmonotonic logics in a single bimodal logic $MBNF$.

Computational mechanisms associated with modal nonmonotonic logics vary greatly in complexity. Because of existence of ranges, there is no simple correlation between the complexity of the underlying modal monotonic logic and the corresponding nonmonotonic modal logic. In the case of logic $S4$ modal nonmonotonic logic is computationally simpler than modal monotonic $S4$ Schwarz and Truszczynski, [70]. The case of autoepistemic logic has been especially thoroughly investigated by Gottlob, [39] and Niemelä, [71]. Similarly to default logic the problems associated with autoepistemic expansions are on the second level of polynomial hierarchy. Specifically, existence problem is Σ_2^P-complete, membership in some expansion is Σ_2^P-complete and membership in all expansion a Π_2^P-complete problem. Generally, the complexity problems for nonmonotonic logics are not completely solved, yet.

7 Circumscription

Circumscription is a second-order technique for minimizing extensions of predicates. In its simplest and most natural form circumscription scheme says that a theory implicitly defines a predicate it circumscribes McCarthy, [72]. Specifically, by circumscribing a predicate P in a theory T we assume that all the information about the predicate is given by T. Thus only those models where P is minimal need to be consider. Formally, let $\varphi = \varphi(P)$ be the conjunction of axioms of a finite theory T in the first order language containing the predicate P. The result of circumscribing P is the following second-order statement

$$\varphi(P) \wedge \neg\exists_Q (Q \subseteq P \wedge \varphi(Q))$$

where Q is a new predicate (not in the language under consideration. We then say that $T_{circ} \models \psi$ if $Circ(T, P) \models \psi$. Clearly, circumscription is syntactically a second-order formula (it quantifies a predicate). In fact for a theory T consisting of these axioms: $\forall_{x,y} R(x, y) \supset P(x, y)$, and $\forall_{x,y,z} R(x, y) \wedge P(y, z) \supset P(x, z)$, the result of circumscribing P by T defines the transitive closure of R – which is not first-order definable, in general. There are cases when circumscription reduces to first-order sentence. For instance theory T consisting of a single axiom $\forall_x Q(x) \wedge R(x) \supset P(x)$ after circumscribing P gives $\forall_x (Q(x) \wedge R(x)) \equiv P(x)$. Lifschitz, in a series of papers [73, 74, 75, 76] analyzed the most important properties of circumscription.

When the theory T is a propositional theory, circumscribing theory T reduces to minimizing models of T (in general we can minimize some or all proposition variables in T). Minimization of all propositional variables at once is known as extended generalized closed world assumption (GCWA) and has been studied by Yahia and Henschen, [77], see also Minker, [78]. Clearly the entailment problem for propositional circumscription is decidable. An algorithm has been proposed by Przymusinski, [79]. In the predicate case the circumscription is very complex. Schlipf, [80] shows that all Δ^1_2 ordinal are definable with circumscription, so this mode of reasoning goes beyond anything currently accepted as computable. The complexity of propositional circumscription has been studied by Eiter and Gottlob, [81] who found that the problems associated with circumscription are, generally, located on the second level of polynomial hierarchy. Circumscription is a very powerful technique and it is known Imielinski, [82] that at least fragment of default logic can be embedded into circumscription.

Suchenek, [83] gave an analysis of various orderings of models and corresponding circumscriptions.

8 Many-valued interpretations

Many-valued logics has been studied in the context of nonmonotonic reasoning, primarily as a technique to get semantics for some logic programs. A four-valued interpretation is a pair $\langle A, B \rangle$ where A, B are two sets of atoms (or elements of Herbrand base of a program). When $A \cap B = \emptyset$, we talk about a three-valued interpretation. 3- and 4- valued logics provide semantics for logic programs. Fitting, [84] noticed that the operator T_P applied to three-valued interpretations is monotone and so has the least fixpoint. He proposed that fixpoint as a semantics for logic programs. A stronger semantics, so called well-founded semantics has been introduced by Van Gelder, Ross and Schlipf, [85]. It turns out that using a three-valued version of stable semantics one can identify well founded semantics with the least stable 3-valued model of the program Przymusinski, [86]. Using the theory of bilattices Ginsberg, [87] Fitting, [88] characterized 3-valued stable models.

Well-founded semantics generalizes perfect semantics for stratified logic programs and in case when it produces a two-valued interpretation the resulting interpretation determines a unique stable model of the program. Van Gelder, [89] gave a polynomial computational procedure for computing well-founded semantics of propositional programs. Schlipf, [90] proved that in the predicate case well-founded semantics can define a complete Π^1_1-set.

9 Truth maintenance and nonmonotonic rule systems

Doyle, [91] considered truth maintenance systems (TMS). These are systems of rules consisting of objects of the following form: $\langle A, B \rangle \to c$ where A, B are finite sets of atoms, c is an atom. The meaning of such rule is "when all the atoms in A are IN and all the atoms in B are OUT then infer c". Doyle did not give a precise semantics for such systems. De Kleer, [92, 93] gave a semantics, very close to supported semantics of logic programs for systems with rules having $B = \emptyset$. Elkan, [94] found a semantics for TMS, essentially the stable semantics of logic programs.

Marek, Nerode and Remmel, [95, 96] considered rules of the form

$$\frac{a_1, \ldots, a_m : b_1, \ldots, b_n}{c} \tag{6}$$

with $a_1, \ldots, a_m, b_1, \ldots, b_n, c$ from an arbitrary set U (not necessarily of atoms) and considered *nonmonotonic rule systems*, i.e. pairs $\langle U, N \rangle$ where U is a set and N consists of rules of the form (6). It turns out that manipulating the parameter U one can provide a proper description of all the first-order forms of nonmonotonic reasoning mentioned above, namely default logic stable and supported semantics of logic programs, McDermott and Doyle systems and truth maintenance systems. Moreover, stripping the systems of specific and unimportant features allows for proving various properties of such systems in a uniform and efficient way.

10 Implementations

As mentioned above there are various algorithms for computations of structures associated with nonmonotonic reasoning. These algorithms have been or presently are being implemented by various groups of researchers. Historically first implementation of (a fragment of) default logic has been reported by Poole, [97]. Dixon and de Kleer, [98] reported massively parallel implementation of TMS. Ginsberg, [99] reported implementation of circumscription. Warren, [100] reported an implementation of well-founded semantics. Bell et al., [101] implemented circumscription and stable semantics for propositional theories using an interpretation of these nonmonotonic systems in linear programming. Niemelä, [102] reported a fast implementation of full autoepistemic logic. Marek and Truszczynski, [103] reported implementation of a default reasoning system.

11 Conclusions and perspectives

Nonmonotonic reasoning is an active area of research in Artificial Intelligence. This research is actively pursued in many places around the world. Many areas not discussed in our short review are currently under development. These include abduction, [104], nonmonotonic probabilistic reasoning, [105], extended logic programming, [106], belief revision, [107] and several other areas. Object-oriented programming, especially its use of classes, inheritance and overriding use technique of inheritance hierarchies with exceptions, [108] and will, doubtless, drive developments in nonmonotonic reasoning. We can expect many discoveries – a definitive description of nonmonotonic reasoning has not been done yet.

References

[1] Aristotle. *Selections.* Oxford University Press, Oxford, 1942.

[2] A. De Morgan. *Formal Logic, or, the calculus of inference, necessary and probable.* Taylor and Walton, London, 1847.

[3] G. Boole. *An investigation of laws of thought, on which are founded mathematical theories of logic and probability.* Macmillan, London, 1854.

[4] C.S. Peirce. *Collected Papers of Charles Sanders Peirce*, volume 3. The Belknap Press of Harvard University Press, Cambridge, MA, 1933. Formal Logic.

[5] G. Frege. *Die Grundlagen der Arithmetic.* M. & H. Marcus, Breslau, 1884.

[6] E. Schröder. *Vorlesungen über die Algebra der Logik.* B.G. Teubner, Leipzig, 1890-1905. 3 vols.

[7] A.N. Whitehead and B. Russel. *Principia Mathematica.* Cambridge University Press, Cambridge, 1913.

[8] D. Hilbert and W. Ackermann. *Principles of Mathematical Logic*. Chelsea, New York, 1950.

[9] K. Gödel. Die Vollständigkeit der Axiome des logische Funktionenkalküls. *Mon. für Math. und Physik*, 37:349–360, 1930.

[10] A. Tarski. *Logic, semantics, metamathematics*. Oxford University Press, Oxford, 1956.

[11] A. Church. A note on Entscheidungsproblem. *Journal of Symbolic Logic*, 1:40–41, 1938.

[12] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Mon. für Math. und Physik*, 38:173–198, 1930.

[13] J. Herbrand. *Logical Writings*. Reidel, Hingham, MA, 1971.

[14] S.C. Kleene. *Introduction to Metamathematics*. D. Van Nostrand, New York, 1952.

[15] A.M. Turing. On computable numbers with an application to the Entscheidungsproblem. *Proceedings of London Mathematical Society*, 2:230–265, 1936.

[16] B.F. Chellas. *Modal logic, an introduction*. Cambridge University Press, 1980.

[17] J. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, Naples, 1983.

[18] A. Colmerauer, H. Kanoui, P. Roussel, and R. Pasero. Un systeme de communication homme-machine en francais. Technical report, Univ. Aix-Marseille, Groupe de Recherche en Intelligence Artificielle, 1973.

[19] T. Coquand and G. Huet. Constructions: a higher-order proof system for mechanizing mathematics. *Springer Lecture Notes in Computer Science*, 242, 1985.

[20] J.-Y. Girard. *Proof and Types*. Cambridge University Press, Cambridge, 1989.

[21] R. L. Constable, S.F. Allen, H.M. Bromley, W.R. Clea veland, J.F. Cremer, R.W. Harper, D.J. Howe, T.B. Knoblock, N.P. Mendler, P. Panangaden, J.T. Sasaki, and S.F. Smith. *Implementing Mathematics with the Nuprl Development System*. Prentice-Hall, NJ, 1986.

[22] P. Martin-Löf. *Intuitionistic Type Theories*. Bibliopolis, Naples, 1984.

[23] J. McCarthy. *Formalization of common sense, papers by John McCarthy edited by V. Lifschitz*. Ablex, 1990.

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

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

[26] D. McDermott and J. Doyle. Nonmonotonic logic I. *Artificial Intelligence*, 13:41–72, 1980.

[27] D. McDermott. Nonmonotonic logic II: nonmonotonic modal theories. *Journal of the ACM*, 29:33–57, 1982.

[28] H. Rasiowa and R. Sikorski. *Mathematics of metamathematics*. Polish Scientific Publishers, PWN, 1970.

[29] J.A. Robinson. Machine-oriented logic based on resolution principle. *Journal of the ACM*, 12:23 – 41, 1965.

[30] R.M. Smullyan. *First-order logic*. Berlin: Springer-Verlag, 1968.

[31] A. Nerode and R. Shore. *Logic and logic programming*. Berlin: Springer-Verlag, 1993.

[32] P. Besnard. *An introduction to default logic*. Springer-Verlag, Berlin, 1989.

[33] D. W. Etherington. *Reasoning with incomplete information*. Pitman, London, 1988.

[34] G. Brewka. Cumulative default logic: in defense of nonmonotonic inference rules. *Artificial Intelligence*, 50:183–205, 1991.

[35] W. Marek and M. Truszczyński. *Nonmonotonic logics; context-dependent reasoning*. Berlin: Springer-Verlag, 1993.

[36] W. Łukaszewicz. *Non-monotonic reasoning: formalization of commonsense reasoning*. Ellis Horwood, 1990.

[37] K. Apt. Logic programming. In J. van Leeuven, editor, *Handbook of theoretical computer science*, pages 493–574. MIT Press, Cambridge, MA, 1990.

[38] F. Yang, H. Blair, and A. Brown. Programming in default logic. University of Syracuse, 1992.

[39] G. Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2:397–425, 1992.

[40] V. Lifschitz. On open defaults. In J. Lloyd, editor, *Proceedings of the symposium on computational logic*, pages 80–95. Berlin: Springer-Verlag, 1990. ESPRIT Basic Research Series.

[41] R. Guerreiro and M. Casanova. An alternative semantics for default logic. Preprint. The Third International Workshop on Nonmonotonic Reasoning, South Lake Tahoe, 1990.

[42] W. Łukaszewicz. Considerations on default logic. In R. Reiter, editor, *Proceedings of the international workshop on non-monotonic reasoning*, pages 165–193, 1984. New Paltz, NY.

[43] C. Baral and V.S. Subrahmanian. Dualities between alternative semantics for logic programming and nonmonotonic reasoning. In A. Nerode, W. Marek, and V.S. Subrahmanian, editors, *Logic programming and non-monotonic reasoning*, pages 69–86. MIT Press, 1991.

[44] W. Marek and M. Truszczyński. Relating autoepistemic and default logics. In *Proceedings of the 1st international conference on principles of knowledge representation and reasoning, KR '89*, pages 276–288, San Mateo, CA., 1989. Morgan Kaufmann.

[45] G. Brewka. *Nonmonotonic reasoning: logical foundations of commonsense.* Cambridge University Press, Cambridge, UK, 1991.

[46] M.H. van Emden and R.A. Kowalski. The semantics of predicate logic as a programming language. *Journal of the ACM*, 23(4):733–742, 1976.

[47] T. Eiter and G. Gottlob. On the complexity of propositional knowledge base revision, updates and counterfactuals. In *ACM Symposium on Principles of Database Systems*, 1992.

[48] K.L. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and data bases*, pages 293–322. Plenum Press, 1978.

[49] K.R. Apt and M.H. van Emden. Contributions to the theory of logic programming. *Journal of the ACM*, 29:841–862, 1982.

[50] K. Apt, H.A. Blair, and A. Walker. Towards a theory of declarative knowledge. In J. Minker, editor, *Foundations of deductive databases and logic programming*, pages 89–142, Los Altos, CA, 1988. Morgan Kaufmann.

[51] K. Apt and H.A. Blair. Arithmetical classification of perfect models of stratified programs. *Fundamenta Informaticae*, 12:1–17, 1990.

[52] T. Przymusiński. On the declarative semantics of deductive databases and logic programs. In *Foundations of deductive databases and logic programming*, pages 193–216, Los Altos, CA., 1988. Morgan Kaufmann.

[53] P. Cholak and H.A. Blair. The complexity of local stratification. *Fundamenta Informaticae*, 1994. To appear.

[54] M. Gelfond and V. Lifschitz. The stable semantics for logic programs. In R. Kowalski and K. Bowen, editors, *Proceedings of the 5th international symposium on logic programming*, pages 1070–1080, Cambridge, MA., 1988. MIT Press.

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

[56] W. Marek and M. Truszczyński. Stable semantics for logic programs and default theories. In E.Lusk and R. Overbeek, editors, *Proceedings of the North American conference on logic programming*, pages 243–256, Cambridge, MA., 1989. MIT Press.

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

[58] W. Marek, A. Nerode, and J. B. Remmel. The stable models of predicate logic programs. In *Proceedings of international joint conference and symposium on logic programming*, Boston, MA, 1992. MIT Press.

[59] R.C. Stalnaker. A note on nonmonotonic modal logic. Unpublished manuscript, 1980.

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

[61] G.F. Shvarts. Stable expansions of autoepistemic theories. In *Proceedings of the all-union conference on artificial intelligence*, pages 483–487, 1988. In Russian.

[62] G.F. Schwarz. Autoepistemic logic of knowledge. In A. Nerode, W. Marek, and V.S. Subrahmanian, editors, *Logic programming and non-monotonic reasoning*, pages 260–274. MIT Press, 1991.

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

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

[65] M. Truszczyński. Modal interpretations of default logic. In *Proceedings of IJCAI-91*, pages 393–398, San Mateo, CA., 1991. Morgan Kaufmann.

[66] M. Gelfond. On stratified autoepistemic theories. In *Proceedings of AAAI-87*, pages 207–211, Los Altos, CA., 1987. American Association for Artificial Intelligence, Morgan Kaufmann.

[67] W. Marek and M. Truszczyński. Reflexive autoepistemic logic and logic programming. In A. Nerode and L. Pereira, editors, *Logic programming and non-monotonic reasoning*. MIT Press, 1993.

[68] 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.

[69] V. Lifschitz. Minimal belief and negation as failure. *Artificial Intelligence*, 1994. To appear.

[70] G.F. Schwarz and M. Truszczyński. Nonmonotonic reasoning is sometimes easier. Technical Report 213-92, Department of Computer Science, University of Kentucky, 1992.

[71] I. Niemelä. On the decidability and complexity of autoepistemic reasoning. *Fundamenta Informaticae*, 17:117–155, 1992.

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

[73] V. Lifschitz. Computing circumscription. In *Proceedings of IJCAI-85*, pages 121–127, Los Altos, CA., 1985. Morgan Kaufmann.

[74] V. Lifschitz. Pointwise circumscription. In M.Ginsberg, editor, *Readings in nonmonotonic reasoning*, pages 179–193, Los Altos, CA., 1987. Morgan Kaufmann.

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

[76] V. Lifschitz. Between circumscription and autoepistemic logic. In *Proceedings of the 1st international conference on principles of knowledge representation and reasoning*, pages 235–244, San Mateo, CA., 1989. Morgan Kaufmann.

[77] A. Yahia and L.J. Henschen. Deduction in non-horn databases. *Journal of Automated Reasoning*, pages 141–160, 1985.

[78] J. Minker. On indefinite databases and the closed world assumption. *Lecture Notes in Computer Science*, 138:292–308, 1982.

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

[80] J. Schlipf. Decidability and definability with circumscription. *Annals of Pure and Applied Logic*, 35:173–191, 1987.

[81] T. Eiter and G. Gottlob. Complexity of reasoning with parsimonious and moderately grounded expansions. *Fundamenta Informaticae*, 1992. to appear.

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

[83] M. Suchenek. Preservation properties in deductive data bases. *Methods of Logic in Computer Science*, 1994. To appear.

[84] M. C. Fitting. Kripke-Kleene semantics for logic programs. *Journal of Logic Programming*, 2:295–312, 1985.

[85] A. Van Gelder, K.A. Ross, and J.S. Schlipf. Unfounded sets and well-founded semantics for general logic programs. In *ACM symposium on principles of database systems*, pages 221–230, 1988.

[86] T. Przymusiński. Three-valued nonmonotonic formalisms and logic programming. In *Proceedings of the 1st international conference on principles of knowledge representation and reasoning, KR '89*, pages 341–348, San Mateo, CA., 1989. Morgan Kaufmann.

[87] M.L. Ginsberg, editor. *Readings in nonmonotonic reasoning.* Morgan Kaufmann, 1987.

[88] M.C. Fitting. Well-founded semantics, generalized. In *Proceedings of the international symposium on logic programming*, pages 71–84, Cambridge, MA., 1991. MIT Press.

[89] A. Van Gelder. The alternating fixpoints of logic programs with negation. In *ACM symposium on principles of database systems*, pages 1– 10, 1989.

[90] J. Schlipf. The expressive powers of the logic programming semantics. *J.C.S.S*, 1994. To appear, A preliminary version appeared in *Ninth ACM Symposium on Principles of Database Systems*, 1990.

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

[92] J. de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

[93] J. de Kleer. Extending the ATMS. *Artificial Intelligence*, 28:163–196, 1986.

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

[95] W. Marek, A. Nerode, and J.B. Remmel. Nonmonotonic rule systems I. *Annals of Mathematics and Artificial Intelligence*, 1:241–273, 1990.

[96] W. Marek, A. Nerode, and J.B. Remmel. Nonmonotonic rule systems II. *Annals of Mathematics and Artificial Intelligence*, 5:229–263, 1992.

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

[98] M. Dixon and J. de Kleer. Massively parallel assumption-based truth maintenance. In M. Reinfrank, J. de Kleer, M.L. Ginsberg, and E. Sandewall, editors, *Non-monotonic reasoning*, pages 131–142. Berlin: Springer-Verlag, 1989. Lecture Notes in Artificial Intelligence, 346.

[99] M.L. Ginsberg. A circumscriptive theorem prover. In M. Reinfrank, J. de Kleer, M.L. Ginsberg, and E. Sandewall, editors, *Non-monotonic reasoning*, pages 100–114. Berlin: Springer-Verlag, 1989. Lecture Notes in Artificial Intelligence, 346.

[100] D.S. Warren W. Chen, T. Swift. Efficient computation of queries under the well-founded semantics. Technical Report 93-CSE-33, Southern Methodist Univeristy, 1993.

[101] C. Bell, A. Nerode, R. Ng, W. Pugh, and V.S. Subrahmanian. Implementing stable semantics by linear programming. In A. Nerode and L. Pereira, editors, *Logic programming and non-monotonic reasoning*. MIT Press, 1993.

[102] I. Niemelä. Autoepistemic logic as a unified basis for nonmonotonic reasoning. Technical Report 24, Helsinki University of Technology, 1993.

[103] W. Marek and M. Truszczyński. Deres – default reasoning system. In preparation, 1994.

[104] A. C. Kakas, R.A. Kowalski, and F. Toni. Abductive logic programming. To appear, 1992.

[105] J. Pearl. *Probabilistic reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[106] M. Gelfond and V. Lifschitz. Logic programs with classical negation. In D. Warren and P. Szeredi, editors, *Proceedings of the 7th international conference on logic programming*, pages 579–597, Cambridge, MA., 1990. MIT Press.

[107] C. E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.

[108] D. Touretzky. *The mathematics of inheritance systems*. Pitman, London, 1986.