Nonmonotonic Databases and Epistemic Queries*

Vladimir Lifschitz
Department of Computer Sciences
and Department of Philosophy
University of Texas at Austin
Austin, TX 78712, U. S. A.

Abstract

The approach to database query evaluation developed by Levesque and Reiter treats databases as first order theories, and *queries* as formulas of the language which includes, in addition to the language of the database, an epistemic modal operator. In this epistemic query language, one can express questions not only about the external world described by the database, but also about the database itself—about what the database knows. On the other hand, epistemic formulas are used in knowledge representation for the purpose of expressing *defaults.* Autoepistemic logic is the best known epistemic nonmonotonic formalism; the logic of grounded knowledge, proposed recently by Lin and Shoham, is another such system. This paper brings these two directions of research together. We describe a new version of the Lin/Shoham logic, similar in spirit to the Levesque/Reiter theory of epistemic queries. Using this formalism, we can give meaning to epistemic queries in the context of nonmonotonic databases, including logic programs with negation as failure.

1 Introduction

The approach to database query evaluation developed by Levesque [1984] and Reiter *[1990]* treats *databases* as first order theories, and *queries* as formulas of the language which includes, in addition to the language of the database, an epistemic modal operator. In this epistemic query language, one can express questions not only about the external world described by the database, but also about the database itself—about what the database knows. For instance, one can ask not only whether John teaches any classes this semester, but also whether there is a known class that John teaches. The first question will be expressed by a formula like *3xTeaches(John, x);* the second, by the epistemic formula *3xK Teaches(John, x).* The difference between these queries becomes essential when the database contains incomplete (for instance, disjunctive) information. Reiter [1988] argues that epistemic formulas are appropriate also for expressing integrity constraints.

On the other hand, epistemic formulas are used in knowledge representation for the purpose of expressing *defaults.* Autoepistemic logic ([Moore, 1985], [Levesque, 1990]) is the best k nown epistemic nonmonotonic formalism. One of the reasons why autoepistemic logic is important is that *general logic programs* can be naturally viewed as autoepistemic theories [Gelfond, 1987]. This is no longer the case, however, for "extended" logic programs, which are capable of handling incomplete information [Gelfond and Lifschitz, 1990]. In order to express extended rules by formulas, one has to use other epistemic nonmonotonic formalisms. The logic of "grounded knowledge," proposed by Lin and Shoham [1990], is one of the possibilities.[1]

This paper brings these two directions of research together. We describe a new version of the Lin/Shoham system, similar in spirit to the Levesque/Reiter theory of epistemic queries. Our formulation is simpler than that of [Lin and Shoham, 1990], and, unlike the latter, it is not restricted to the propositional case. Using this formalism, we can give meaning to epistemic queries in the context of logic programming; we can ask what a *logic program knows.* Because our system contains also (some forms of) default logic [Reiter, 1980] and circumscription [McCarthy, 1986], we can give meaning to epistemic queries in the context of a default theory or a circumscriptive theory as well.

The system of Lin and Shoham, unlike most other epistemic nonmonotonic formalisms, uses *two* epistemic operators. One of them represents "minimal knowledge"[2]; the other is closely related to the concepts of "justification" in default logic and of "negation as failure" in logic programming. The main technical idea of this paper is to identify the former with the epistemic operator K used by Levesque and Reiter.

'This research was supported in part by NSF grant IRI-89-04611 and by DARPA under Contract N00039-84-C-02n.

[1] Other modifications of autoepistemic logic that can be used for this purpose were developed in [Marek and Truszczyriski, 1989], [Siegel, 1990] and [Truszczyriski, 1991].

[2] The idea of minimal knowledge (or "maximal ignorance") was formalized, in various ways, by several authors, including Konolige [1982], Halpern and Moses [1984], Shoham [1986] and Lin [1988].

We concentrate here entirely on the semantical issues, and leave aside, for the time being, the computational problems of query evaluation in this framework.[3]

2 Propositional MKNF: Formulas and Interpretations

The formulas of the propositional logic of minimal knowledge with negation as failure (MKNF) are built from propositional symbols (atoms) using the standard propositional connectives and two modal operators: K and *not*.[4] A *theory* is a set of formulas (*axioms*).

If a formula or a theory doesn't contain the negation as failure operator *not*, we call it *positive*. If it contains neither K nor *not*, it is *objective*.

An *interpretation* is a set of atoms. The set of all interpretations will be denoted by $\mathcal{I}$. Our first goal is to define when a set $S \subset \mathcal{I}$ is a model of a theory T. As a preliminary step, let us consider the case of positive theories.

3 Positive Theories

For an interpretation I and a set of interpretations S, we define when a positive formula F is *true* in (I, S), as follows. (For simplicity, we assume that all propositional connectives are expressed in terms of the primitives $\neg$ and $\wedge$.)

1. If F is an atom, F is true in (I, S) iff $F \in I$.
2. $\neg F$ is true in (I, S) iff F is not true in (I, S).
3. $F \wedge G$ is true in (I, S) iff F and G are both true in (I, S).
4. KF is true in (I, S) iff, for every $J \in S$, F is true in (J, S).

A model of a positive theory T is any maximal set $S \subset \mathcal{I}$ such that, for every $I \in S$, the axioms of T are true in (I, S).

The maximality condition expresses the idea of "minimal knowledge": The larger the set of "possible worlds" is, the fewer facts are known.

In the special case when T is objective, the requirement that all axioms of T be true in (I, S) means simply that I is a model of T in the sense of classical propositional logic. Consequently, the set *Mod*(T) of the "classical models" of T is the only model of T in the sense of MKNF.

As another example, consider the case when T is $\{KF\}$, where F is an objective formula. The models of T are the maximal sets $S \subset \mathcal{I}$ such that every interpretation from S is a classical model of F. It is clear that *Mod*(F) is the only such set. More generally, if F is an objective formula and T_0 a set of objective formulas, then $T_0 \cup \{KF\}$ has a unique model; it is the same as the model of $T_0 \cup \{F\}$, that is, *Mod*$(T_0 \cup \{F\})$.

If $T = \{KF_1 \vee KF_2\}$, where F_1, F_2 are objective, then the models of T are the maximal sets $S \subset \mathcal{I}$ that satisfy the condition: Every interpretation from S is a classical model of F_1 or every interpretation from S is a classical model of F_2; symbolically:

$$S \subset Mod(F_1) \text{ or } S \subset Mod(F_2).$$

If neither F_1 nor F_2 is a logical consequence of the other, then neither of the sets *Mod*(F_1), *Mod*(F_2) contains the other, and T has two models: *Mod*(F_1) and *Mod*(F_2).

4 Propositional MKNF: Models

How can we extend the definition of a model to the general case, when the axioms contain *not*?

In the presence of both K and *not*, truth will be defined relative to a triple (I, S^k, S^n), where S^k and S^n are sets of interpretations; S^k serves as the set of possible worlds for the purpose of defining the meaning of K, and S^n plays the same role for the operator *not*. Then a model will be defined as any set S^n that satisfies a certain fixpoint condition.

For an interpretation I and two sets of interpretations S^k, S^n, we define when a formula is F *true* in (I, S^k, S^n), as follows.

1. If F is an atom, F is true in (I, S^k, S^n) iff $F \in I$.
2. $\neg F$ is true in (I, S^k, S^n) iff F is not true in (I, S^k, S^n).
3. $F \wedge G$ is true in (I, S^k, S^n) iff F and G are both true in (I, S^k, S^n).
4. KF is true in (I, S^k, S^n) iff, for every $J \in S^k$, F is true in (J, S^k, S^n).
5. *not* F is true in (I, S^k, S^n) iff, for some $J \in S^n$, F is not true in (J, S^k, S^n).

The truth condition for *not* F expresses that F is not known to be true provided that S^n is the set of worlds that are considered "possible."

This definition is a generalization of the definition of truth for positive formulas given in the previous section, in the sense that a positive formula is true in (I, S^k, S^n) iff it is true in (I, S^k).

For any theory T and any set $S \subset \mathcal{I}$, by $\Gamma(T, S)$ we denote the set of all maximal sets $S' \subset \mathcal{I}$ which satisfy the condition:

$$\text{For every } I \in S', \text{ the axioms of } T \text{ are true in } (I, S', S). \quad (1)$$

If $S \in \Gamma(T, S)$, then we say that S is a *model* of T.

It is easy to see that, for positive theories, this is equivalent to the definition given earlier. Indeed, if T is positive, then $\Gamma(T, S)$ does not depend on S, and is simply the set of all models of T in the sense defined in the previous section.

As an example of a theory whose axioms are not positive, consider the theory T whose only axiom is

$$not\ F \supset G, \quad (2)$$

where F and G are objective. The condition (1) says in this case:

$$\text{For every } I \in S', \text{ (2) is true in } (I, S', S).$$

This is equivalent to:

[3] An algorithm for the evaluation of epistemic queries in the monotonic case is proposed in [Reiter, 1990].

[4] The "assumption operator" A from [Lin and Shoham, 1990]) corresponds, in our notation, to the combination $\neg$*not*.

$$\text{If, for some } J \in S,\ F \text{ is false in } J,$$
$$\text{then, for every } I \in S',\ G \text{ is true in } I,$$

or

$$\text{If } S \not\subset Mod(F) \text{ then } S' \subset Mod(G).$$

Consequently $\Gamma(T, S)$ is $\{\mathcal{I}\}$ if $S \subset Mod(F)$, and $\{Mod(G)\}$ otherwise. We conclude that the models of (2) can be characterized as follows:

1. If F is a tautology, then $\mathcal{I}$ is the only model of (2).
2. If F is a not a tautology but is a logical consequence of G, then (2) has no models. For instance, *not* $F \supset F$ has no models.
3. If F is a not a logical consequence of G, then the only model of (2) is $Mod(G)$. For instance, *not* $p \supset q$, where p and q are distinct atoms, has one model, $Mod(q)$.

5 Relation to Logic Programming

In this section we show that logic programs of some kinds can be viewed as theories in the sense of MKNF. We will consider three classes of programs, moving gradually towards greater generality. In the semantics of logic programming, it is customary to view a rule with variables as shorthand for the set of its ground instances; for this reason, we can restrict our attention to propositional programs.

A *positive logic program* is a set of rules of the form

$$A_0 \leftarrow A_1, \ldots, A_m, \tag{3}$$

where $m \geq 0$, and each A_i is an atom. According to van Emden and Kowalski [1976], the semantics of a positive program Π is defined by the smallest set of atoms which is closed under its rules (that is to say, which includes A_0 whenever it includes $A_1, \ldots, A_m$, for every rule (3) from Π). This set of atoms is known as the "minimal model" of Π, and we will denote it by $MM(\Pi)$.

In order to relate the semantics of positive programs to MKNF, let us agree to identify a rule (3) with the formula

$$\mathrm{K}A_1 \wedge \ldots \wedge \mathrm{K}A_m \supset \mathrm{K}A_0.$$

For any interpretation J, let $\omega(J)$ be the set of all interpretations I such that $J \subset I$.

Theorem 1, Part A. *Every positive program Π has one model, $\omega(MM(\Pi))$.*

For instance, let the rules of Π be

$$p \leftarrow q;\ r \leftarrow s;\ s \leftarrow . \tag{4}$$

Then $MM(\Pi)$ is $\{r, s\}$. Viewed as a theory in the sense of MKNF, Π is the set of 3 axioms:

$$\mathrm{K}q \supset \mathrm{K}p,\ \mathrm{K}s \supset \mathrm{K}r,\ \mathrm{K}s;$$

its model $\omega(MM(\Pi))$ is

$$\{\{r, s\}, \{p, r, s\}, \{q, r, s\}, \{p, q, r, s\}\}. \tag{5}$$

A *general logic program* is a set of rules of the form

$$A_0 \leftarrow A_1, \ldots, A_m, not\ A_{m+1}, \ldots, not\ A_n, \tag{6}$$

where $n \geq m \geq 0$, and each A_i is an atom. Several approaches to defining a semantics for general logic programs have been proposed. One of them is based on the notion of a "stable model" [Gelfond and Lifschitz, 1988]. A stable model of Π is an interpretation satisfying some fixpoint condition. We will denote the set of such interpretations by $SM(\Pi)$. It is clear from the definition of a stable model that, in the special case when Π is positive, $SM(\Pi) = \{MM(\Pi)\}$.

A rule (6) will be identified with the formula

$$\mathrm{K}A_1 \wedge \ldots \wedge \mathrm{K}A_m \wedge not\ A_{m+1} \wedge \ldots \wedge not\ A_n \supset \mathrm{K}A_0.$$

Theorem 1, Part B. *The set of models of a general program Π is*

$$\{\omega(M)\ :\ M \in SM(\Pi)\}.$$

For instance, the program with the rules

$$p \leftarrow not\ q;\ q \leftarrow not\ r$$

has one stable model, $\{q\}$. These rules can be written as

$$not\ q \supset \mathrm{K}p,\ not\ r \supset \mathrm{K}q.$$

The only model of these axioms is

$$\{\{q\}, \{p, q\}, \{q, r\}, \{p, q, r\}\}.$$

Finally, we will consider the class of (extended) disjunctive databases in the sense of [Gelfond and Lifschitz, 1991], in which classical negation and a form of disjunction are allowed. A *disjunctive database* is a set of rules of the form

$$L_1 \mid \ldots \mid L_l \leftarrow L_{l+1}, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n, \tag{7}$$

where $n \geq m \geq l \geq 0$, and each L_i is a literal (an atom possibly preceded by $\neg$). The semantics of disjunctive databases defines when a set of literals is an "answer set" of a given database. We will denote the set of answer sets of a disjunctive database Π by $AS(\Pi)$. If Π is a general logic program, then $AS(\Pi) = SM(\Pi)$.

A rule (7) will be identified with the formula

$$\mathrm{K}L_{l+1} \wedge \ldots \wedge \mathrm{K}L_m \wedge not\ L_{m+1} \wedge \ldots \wedge not\ L_n \supset \mathrm{K}L_1 \vee \ldots \vee \mathrm{K}L_l. \tag{8}$$

In order to extend Theorem 1 to disjunctive databases, we need to generalize the definition of the operator ω. So far, $\omega(M)$ is defined when M is an interpretation, that is, a set of atoms. Now M is allowed to be a set of literals. For any set of literals M, M^+ is the set of atoms that belong to M, and M^- the set of atoms whose negations belong to M; $\omega(M)$ stands for the set of interpretations I such that $M^+ \subset I$ and $M^- \cap I = \emptyset$.

Theorem 1, Part C. *The set of models of a disjunctive database Π is*

$$\{\omega(M)\ :\ M \in AS(\Pi)\}.$$

For instance, the disjunctive database whose only rule is

$$p \mid \neg q \leftarrow not\ r \tag{9}$$

has two answer sets: $\{p\}$ and $\{\neg q\}$. This rule is the same as the formula

$$not\ r \supset \mathrm{K}p \vee \mathrm{K}\neg q;$$

the models of this axiom are

$$\{\{p\}, \{p, q\}, \{p, r\}, \{p, q, r\}\}$$

and

$$\{\emptyset, \{p\}, \{r\}, \{p, r\}\}.$$

6 Propositional MKNF: The Consequence Relation

We say that a positive formula F is a *theorem* of a theory T, or a *consequence* of its axioms (symbolically, $T \models F$), if, for every model S of T and every interpretation $I \in S$, F is true in (I, S). Thus theoremhood is defined for positive formulas only. (We do not see any reasonable way to define theoremhood for formulas containing *not*.)

To illustrate this definition, consider first the special case when all axioms of T are objective. If F is objective also, then $T \models F$ if and only if F is a consequence of T in the sense of classical propositional logic. If F is allowed to contain K, then the definition of $\models$ turns into (the propositional case of) the definition of $\approx$ from [Reiter, 1990].

If, for instance, the only axiom of T is $p \vee q$, then $\neg Kp$ is a theorem. If $\neg q$ is added as another axiom, this theorem will be lost. Thus the consequence relation $\models$ is nonmonotonic even for objective T.

If Π is a general logic program or a disjunctive database, we can ask whether a positive formula F is a consequence of Π. For instance, $q \supset p$ is not a consequence of the program (4), because it is false in one of the interpretations from (5). Theorem 1 (Part C) shows that a literal is a consequence of Π if and only if it belongs to all answer sets of Π.

We conclude this section with two remarks about general properties of the consequence relation. First, we would like to say that every axiom is a theorem, but this makes sense for positive axioms only. It is easy to check, however, that the result of replacing all occurences of *not* by $\neg$K in an axiom is a theorem. Second, let F_1 and F_2 be positive formulas; if $KF_1 \supset KF_2$ is provable in $S5$, and $T \models F_1$, then $T \models F_2$. In this sense, the set of theorems is closed under $S5$.

7 Quantification

Our next goal is to extend the definitions given above to languages with quantification. For simplicity, we consider first-order quantifiers only; extension to the higher-order case is straightforward.

Consider the language obtained from a first-order language $\mathcal{L}$ by adding the modal operators K and *not*. A *theory* is now a set of sentences of this language; an *interpretation* is a structure for $\mathcal{L}$. The universe of an interpretation I will be denoted by $|I|$. For any nonempty set U, by $\mathcal{I}_U$ we denote the set of all interpretations with the universe U.

Let I be an interpretation, and let S^k, S^n be subsets of $\mathcal{I}_{|I|}$. We will define when a sentence is *true* in (I, S^k, S^n). To this end, we need to extend the language by object constants representing all elements of $|I|$; these constants will be called *names*. Truth will be defined for all sentences of the extended language. We assume that all propositional connectives and quantifiers are expressed in terms of $\neg$, $\wedge$ and $\forall$.

1. If F is an atomic sentence, F is true in (I, S^k, S^n) iff F is true in I.
2. $\neg F$ is true in (I, S^k, S^n) iff F is not true in (I, S^k, S^n).
3. $F \wedge G$ is true in (I, S^k, S^n) iff F and G are both true in (I, S^k, S^n).
4. $\forall x F(x)$ is true in (I, S^k, S^n) iff, for every name ξ, $F(\xi)$ is true in (I, S^k, S^n).
5. KF is true in (I, S^k, S^n) iff, for every $J \in S^k$, F is true in (J, S^k, S^n).
6. *not* F is true in (I, S^k, S^n) iff, for some $J \in S^n$, F is not true in (J, S^k, S^n).

For any theory T, nonempty set U, and subset S of $\mathcal{I}_U$, by $\Gamma_U(T, S)$ we denote the set of all maximal sets $S' \subset \mathcal{I}_U$ which satisfy the condition:

For every $I \in S'$, the axioms of T are true in (I, S', S).

If $S \in \Gamma_U(T, S)$, then we say that (U, S) is a *model* of T.

If all axioms of T are objective, then the models of T are the pairs $(U, Mod_U(T))$, where $Mod_U(T)$ stands for the set of classical models of T with the universe U.

With quantification available, we can represent logic programs as axiom sets in a more direct way, without first replacing rules by their ground instances. A disjunctive database can be identified with the axiom set consisting of (i) the universal closures of the formulas (8) corresponding to its rules (7), and (ii) appropriate equality axioms (see [Clark, 1978]). This semantics differs from the one presented in Section 5 in that it permits "non-Herbrand models."[5]

A positive sentence F is a *theorem* of a theory T if, for every model (U, S) of T and every interpretation $I \in S$, F is true in (I, S). The two properties of the propositional consequence relation stated at the end of the previous section apply to the first-order case also.

8 Relation to Default Logic and Circumscription

In accordance with the idea of Lin and Shoham [1990], we identify a default

$$\alpha : \beta_1, \ldots, \beta_m / \gamma \tag{10}$$

with (the universal closure of) the formula

$$K\alpha \wedge not \neg\beta_1 \wedge \ldots \wedge not \neg\beta_m \supset K\gamma. \tag{11}$$

The following theorem refers to "default logic with a fixed universe"—a modification of the system of [Reiter, 1980] proposed in [Lifschitz, 1990]. The main difference is that, in the standard default logic, parameters of an open default are treated as metavariables for ground terms, whereas the modified system handles parameters as genuine object variables.

Theorem 2. *An objective sentence F is a fixed-universe consequence of (D, W) iff $D \cup W \models F$.*

This theorem shows that the translation (11) embeds default logic with a fixed universe into MKNF.

In [Lifschitz, 1990] we showed how to embed circumscription (with all nonlogical constants varied) into default logic with a fixed universe. The composition of

[5] See [Przymusinski, 1989] on the role of non-Herbrand models in logic programming.

these two transformations reduces the circumscription of P in a sentence $F(P)$ to the formula

$$F(P) \wedge \forall x(not\ P(x) \supset \neg P(x)).$$

The objective consequences of this formula in the sense of MKNF are exactly the sentences that follow from the circumscription in classical logic.

9 Conclusion

The logic of minimal knowledge with negation as failure provides a unified framework for several nonmonotonic formalisms and for the Levesque/Reiter theory of epistemic queries. Its semantics, like the semantics of circumscription and of default logic with a fixed universe, is a generalization of the standard concept of a model of a first order theory; we consider this an important advantage.

However, this unification is not entirely satisfactory, for two reasons. First, the logic of minimal knowledge (even in the propositional case and without negation as failure) has the following puzzling and unintuitive property[6] When a theory T is extended by an "explicit definition'* of an atom p—by an axiom p = F, where p occurs neither in the axioms of T nor in F—this may affect the class of theorems that do not contain p. In other words, in the logic of minimal knowledge, a "definitional" extension is not necessarily "conservative." This observation seems to point to a serious defect of the idea of minimal knowledge. Second, MKNF does not cover the important concept of "strong introspection," introduced recently by Gelfond [1991].

Acknowledgments

I would like to thank Michael Gelfond, Katsumi Inoue, Hector Levesque, Fangzhen Lin, Ray Reiter, Grigorii Schwarz and Miroslaw Truszezyhski for useful discussions on the subject of this paper.

References

[Clark, 1978] Keith Clark. Negation as failure. In Herve Gallaire and Jack Minker, editors, *Logic and Data Bases,* pages 293-322. Plenum Press, New York, 1978.

[Emden and Kowalski, 1976] Maarten van Emden and Robert Kowalski. The semantics of predicate logic as a programming language. *Journal of the ACM,* 23(4)733-742, 1976.

[Gelfond and Lifschitz, 1988] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In Robert Kowalski and Kenneth Bowen, editors, *Logic Programming: Proc. of the Fifth InVl Conf and Symp.,* pages 1070-1080, 1988.

[Gelfond and Lifschitz, 1990] Michael Gelfond and Vladimir Lifschitz. Logic programs with classical negation. In David Warren and Peter Szeredi, editors, *Logic Programming: Proc. of the Seventh Int'l Conf,* pages 579-597, 1990,

[6]Grigorii Schwarz, personal communication.

[Gelfond and Lifschitz, 1991] Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing,* 1991. To apppear.

[Gelfond, 1987] Michael Gelfond. On stratified autoepistemic theories. In *Proc. AAAI-87,* pages 207-211, 1987.

[Gelfond, 1991] Michael Gelfond. Strong introspection. In *Proc. AAAI-91,* 1991. To apppear.

[Halpern and Moses, 1984] Joseph Halpern and Yoram Moses. Towards a theory of knowledge and ignorance: preliminary report. Technical Report RJ 4448 (48136), IBM, 1984.

[Konolige, 1982] Kurt Konolige. Circumscriptive ignorance. In *Proc. of AAAI-82,* pages 202-204, 1982,

[Levesque, 1984] Hector Levesque. Foundations of a functional approach to knowledge representation. *Artificial Intelligence,* 23(2): 155-212, 1984.

[Levesque, 1990] Hector Levesque. All I know; a study in autoepistemic logic. *Artificial Intelligence,* 42(2,3):263-310, 1990.

[Lifschitz, 1990] Vladimir Lifschitz. On open defaults. In John Lloyd, editor, *Computational Logic: Symposium Proceedings,* pages 80-95. Springer, 1990.

[Lin and Shoham, 1990] Fang2hen Lin and Yoav Shoham. Epistemic semantics for fixed-points nonmonotonic logics. In Rohit Parikh, editor. *Theoretical Aspects of Reasoning about Knowledge: Proc, of the Third Conf,* pages 111-120, 1990.

[Lin, 1988] Fangzhen Lin. Circumscription in a modal logic. In Moshe Vardi, editor, *Theoretical Aspects of Reasoning about Knowledge: Proc. of the Second Conf,* pages 113-127, 1988.

[Marek and Truszczyriski, 1989] Wiktor Marek and Miroslaw Truszczynski. Relating autoepistemic and default logic. In Ronald Brachman, Hector Levesque, and Raymond Reiter, editors, *Proc. of the First Int 7 Conf on Principles of Knowledge Representation and Reasoning,* pages 276-288, *1989.*

[McCarthy, 1986] John McCarthy. Applicat ions of circumscription to formalizing common sense knowledge. *Artificial Intelligence,* 26(3):89-l 16, 1986.

[Moore, 1985] Robert Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence,* 25(l):75-94, 1985.

[Przymusjnski, *1989] Teodor* Przymusinski. On the declarative and procedural semantics of logic programs. *Journal of Automated Reasoning,* 5:167-205, 1989.

[Reiter, 1980] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence,* 13(1.2):81-132, 1980.

[Reiter, 1988] Raymond Reiter, On integrity constraints. In Moshe Vardi, editor, *Theoretical Aspects of Reasoning about Knowledge: Proc. of the Second Conf,* pages 97-111, 1988, [Reiter, 1990] Raymond Reiter, On asking what a database knows. In John Lloyd, editor, *Computational Logic: Symposium Proceedings,* pages 96-113. Springer, 1990.

[Shoham, 1986] Yoav Shoham, Chronological ignorance: Time, nonmonotonicity, necessity and causal theories. In *Proc. of AAAI-86,* pages 389-393, 1986.

[Siegel, 1990] Pierre Siegel. A modal language for nonmonotonic logic. Presented at the DRUMS/CEE Workshop, Marseille, 1990.

[Truszczynski, 1991] Miroslaw Truszczynski. Modal interpretations of default logic. In *Proc. of IJCA1-91,* 1991.