Minimal Belief and Negation as Failure*

Vladimir Lifschitz
Department of Computer Sciences
and Department of Philosophy
University of Texas at Austin
Austin, TX 78712, USA

Abstract

Fangzhen Lin and Yoav Shoham defined a propositional nonmonotonic logic which uses two independent modal operators. One of them represents minimal knowledge, the other is related to the ideas of justification (as understood in default logic) and of negation as failure. We describe a simplified version of that system, show how quantifiers can be included in it, and study its relation to circumscription and default logic, to logic programming, and to the theory of epistemic queries developed by Hector Levesque and Ray Reiter.

1 Introduction

Lin and Shoham [16] defined a propositional nonmonotonic logic which uses two independent modal operators. One of them represents minimal knowledge,[1] the other is related to the ideas of justification (as understood in default logic) and of negation as failure. In this paper, we consider a special case of that system, in which Kripke structures of a particularly simple kind are used, and show how quantifiers can be included in it. This extension is based on the ideas of the theory of epistemic

*This research was supported in part by NSF grants IRI-89-04611 and IRI-91-01078, and by DARPA under Contract N00039-84-C-0211.

[1]The idea of minimal knowledge (or "maximal ignorance") was formalized earlier, in various ways, by several authors, including Konolige [8], Halpern and Moses [6], Shoham [19] and Lin [15].

queries developed by Levesque [9] and Reiter [18]. We prove that this system, *the logic of minimal belief and negation as failure (MBNF)*, includes some forms of default logic and circumscription, as well as some logic programming languages. This work can be viewed as an extension of the Levesque/Reiter theory of epistemic queries to nonmonotonic databases and logic programs.

The nonmonotonic logic described here is very expressive, and, accordingly, very intractable. We do not propose to use it directly for representing knowledge; its relatively restricted subsystems, such as languages of logic programming, are more appropriate for this purpose. Being a common framework that unifies several nonmonotonic formalisms, MBNF may help us better understand the possibilities and limitations of each of them.

The uses of MBNF described in the literature so far have to do with the semantics of logic programming. In [14], *MBNF theories with protected literals* are defined—theories that are more general than disjunctive logic programs in view of the possibility of positive occurrences of the negation as failure operator. Inoue and Sakama [7] show that such generalized disjunctive rules are important for abductive logic programming. MBNF was also used in recent work on the relation of logic programming to the logic of "only knowing" [1] and autoepistemic logic [13].

We first discuss the propositional case of MBNF and show how logic programs can be viewed as propositional MBNF theories. Then quantification is added, and connections with default logic and circumscription are investigated. Then databases and answers to queries are defined. Proofs of theorems are given in the appendix.

This paper is a revision of the preliminary report [12]. The system described here is somewhat different from the preliminary version; it avoids the problem with definitional extensions mentioned in the concluding section of [12].

2 Propositional MBNF: Formulas and Interpretations

The formulas of the propositional logic of minimal belief and negation as failure are built from propositional symbols (atoms) using the standard propositional connectives and two modal operators: B and *not*.[2] A *theory* is a set of formulas (*axioms*).

If a formula or a theory does not contain the negation as failure operator *not*, we call it *positive*. If it contains neither B nor *not*, it is *nonmodal*.

[2]The "knowledge operator" K from [16] corresponds to our B; the "assumption operator" A corresponds, in our notation, to the combination $\neg not$.

An *interpretation* is a set of atoms. A *structure* is a pair (I, S), where I is an interpretation, and S a set of interpretations. Intuitively, I represents "the real world," and S the set of "possible worlds." Notice that I is not assumed to be an element of S, so that structures in this sense model belief (possibly unsound), rather than knowledge.

Our first goal is to define when a structure (I, S) is a model of a theory T. As a preliminary step, let us consider the case of positive theories.

3 Positive Theories

We define when a positive formula F is *true* in a structure (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. $\mathsf{B}F$ is true in (I, S) iff, for every $J \in S$, F is true in (J, S).

A structure (I, S) is a *model* of a positive theory T if (i) the axioms of T are true in (I, S), and (ii) there is no structure (I', S') such that S' is a proper superset of S and the axioms of T are true in (I', S').

The maximality of S, required by condition (ii), expresses the idea of "minimal belief": The larger the set of "possible worlds" is, the fewer propositions are believed.

In the special case when T is nonmodal, there is a simple 1–1 correspondence between the models of T in the sense of MBNF and the models of T in the sense of classical logic. Indeed, for a nonmodal theory T, the requirement that the axioms of T be true in (I, S) means simply that I is a "classical model" of T. We will denote the set of models of a nonmodal formula F in the sense of classical logic by $Mod(F)$, and similarly for nonmodal theories. Then the models of a nonmodal theory T in the sense of MBNF are the pairs (I, S), where $I \in Mod(F)$ and S is the set of all interpretations.

As another example, consider the case when T is $\{\mathsf{B}F\}$, where F is nonmodal. Clearly, $\mathsf{B}F$ is true in a structure (I, S) iff S is contained in $Mod(F)$. Consequently, the models of $\{\mathsf{B}F\}$ in the sense of MBNF are the structures of the form $(I, Mod(F))$. More generally, for any theory T, define

$$\mathsf{B}T = \{\mathsf{B}F : F \in T\}.$$

If T is nonmodal, then the models of $\mathsf{B}T$ are the structures of the form $(I, Mod(T))$.

Now let T be $\{\mathsf{B}F_1 \vee \mathsf{B}F_2\}$, where F_1, F_2 are nonmodal formulas. This disjunction is true in (I, S) iff $S \subset Mod(F_1)$ or $S \subset Mod(F_2)$.[3] If neither of the formulas F_1, F_2 is a logical consequence of the other, then neither of the sets $Mod(F_1)$, $Mod(F_2)$ contains the other, and the models of T are the structures of the forms $(I, Mod(F_1))$ and $(I, Mod(F_2))$.

4 Propositional MBNF: Models

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

In the presence of both B and *not*, truth will be defined relative to a triple (I, S^b, S^n), where S^b and S^n are sets of interpretations; S^b serves as the set of "possible worlds" for the purpose of defining the meaning of B, and S^n plays the same role for the operator *not*.

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

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

The truth condition for *not* F expresses that F is not necessarily 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^b, S^n) iff it is true in (I, S^b).

A structure (I, S) is a *model* of a theory T if (i) the axioms of T are true in (I, S, S), and (ii) there is no structure (I', S') such that S' is a proper superset of S and the axioms of T are true in (I', S', S).

It is easy to see that, for positive theories, this is equivalent to the definition given earlier. As an example of a theory whose axioms are not positive, consider the theory T whose only axiom is

$$\textit{not}\ F \supset \mathsf{B}G, \tag{1}$$

where F and G are nonmodal. Formulas of this form are interesting in view of their connection with logic programming. We will see in the next section that, for atomic

[3]We write $A \subset B$ if A is a subset of B, not necessarily proper.

F and G, (1) can be identified with the rule

$$G \leftarrow \mathit{not}\ F.$$

The description of the models of (1) below can be viewed as the analysis of a possible meaning of this rule with F and G not necessarily atomic.

It is clear that (1) is true in (I', S', S) when the following condition is satisfied: If, for some $J \in S$, F is false in J, then, for every $J' \in S'$, G is true in J'. This is equivalent to the condition: $S \subset \mathit{Mod}(F)$ or $S' \subset \mathit{Mod}(G)$. Consequently, the two conditions in the definition of a model can be stated in this case as follows: (i) $S \subset \mathit{Mod}(F)$ or $S \subset \mathit{Mod}(G)$; (ii) there is no proper superset S' of S such that $S \subset \mathit{Mod}(F)$ or $S' \subset \mathit{Mod}(G)$. Consider three cases.

Case 1: F is a tautology. Then condition (i) is trivial, and condition (ii) expresses that S is the set of all interpretations. Consequently, the models of (1) are the structures (I, S) in which S is the set of all interpretations.

Case 2: F is a not a tautology, but it is a logical consequence of G, so that $\mathit{Mod}(G) \subset \mathit{Mod}(F)$. Then condition (i) turns into $S \subset \mathit{Mod}(F)$, and condition (ii) is impossible to satisfy: one can always take S' to be the set of all interpretations. Consequently, (1) has no models. For instance, $\mathit{not}\ p \supset \mathsf{B}p$, where p is an atom, has no models.

Case 3: F is a not a logical consequence of G. Then $S \not\subset \mathit{Mod}(F)$, because otherwise condition (ii) would be impossible to satisfy, as in the previous case. Consequently, condition (i) turns into $S \subset \mathit{Mod}(G)$, and condition (ii) says that S is a maximal set with this property, that is, $S = \mathit{Mod}(G)$. The models of (1) are the structures of the form $(I, \mathit{Mod}(G))$. For instance, the models of $\mathit{not}\ p \supset \mathsf{B}q$, where p and q are distinct atoms, are the structures of the form $(I, \mathit{Mod}(q))$.

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 MBNF. 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

$$A_0 \leftarrow A_1, \ldots, A_m, \tag{2}$$

where $m \geq 0$, and each A_i is an atom. According to van Emden and Kowalski [2], 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 (2) from Π). This set of atoms is known as the "minimal model" of Π. Strictly speaking, this use of the word "model" is only appropriate if one does not distinguish between $\leftarrow$ and material implication, and (2) is identified with a Horn clause. We accept here a different convention: rule (2) will be identified with the formula

$$\mathsf{B}A_1 \wedge \ldots \wedge \mathsf{B}A_m \supset \mathsf{B}A_0.$$

Theorem 1, Part A. *The models of a positive program* Π *are the structures of the form* $(I, Mod(M))$*, where* M *is the "minimal model" of* Π.

For instance, let the rules of Π be

$$p \leftarrow q;\ r \leftarrow s;\ s \leftarrow . \tag{3}$$

Then the "minimal model" of Π is $\{r, s\}$. Viewed as a theory in the sense of MBNF, Π is the set of three axioms:

$$\mathsf{B}q \supset \mathsf{B}p,\ \mathsf{B}s \supset \mathsf{B}r,\ \mathsf{B}s.$$

Its models are the structures of the form

$$(I, \{I' : r, s \in I'\}). \tag{4}$$

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

$$A_0 \leftarrow A_1, \ldots, A_m, not\ A_{m+1}, \ldots, not\ A_n, \tag{5}$$

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" [4]. The use of the word "model" here is rather unfortunate: it would only make sense if we were willing to identify $\leftarrow$ with material implication, and *not* with classical negation. Our convention will be to identify rule (5) with the formula

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

Theorem 1, Part B. *The models of a general program* Π *are the structures of the form* $(I, Mod(M))$*, where* M *is a "stable model" of* Π.

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

$$not\ q \supset \mathsf{B}p,\ \ not\ r \supset \mathsf{B}q.$$

The models of these axioms are the structures of the form

$$(I, \{I' : q \in I'\}).$$

Part B of Theorem 1 is a generalization of Part A, because, for a positive program, its "minimal model" is its only "stable model."

Finally, we will consider the class of "disjunctive" logic programs, in which classical negation and a form of disjunction are allowed. Such programs are introduced in [5] (under the name of "extended disjunctive databases").

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

$$L_1 \mid \ldots \mid L_l \leftarrow L_{l+1}, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n, \tag{6}$$

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 programs defines when a set of literals is an "answer set" of a given program.[4] If Π is a general program, then its answer sets are identical to its "stable models."

A rule (6) will be identified with the formula

$$\mathsf{B}L_{l+1} \wedge \ldots \wedge \mathsf{B}L_m \wedge not\ L_{m+1} \wedge \ldots \wedge not\ L_n \supset \mathsf{B}L_1 \vee \ldots \vee \mathsf{B}L_l. \tag{7}$$

Theorem 1, Part C. *The models of a disjunctive program* Π *are the structures of the form* $(I, Mod(M))$*, where* M *is an answer set of* Π.

For instance, the disjunctive program whose only rule is

$$p \mid \neg q \leftarrow not\ r$$

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

$$not\ r \supset \mathsf{B}p \vee \mathsf{B}\neg q;$$

the models of this axiom are the structures

$$(I, \{I' : p \in I'\})$$

and

$$(I, \{I' : q \notin I'\}).$$

[4]The definition of an answer set is reproduced in the appendix.

6 Propositional MBNF: The Consequence Relation

We say that a positive formula F is a *theorem* of a theory T, or is *entailed* by its axioms (symbolically, $T \models F$), if F is true in every model of T. Thus theoremhood is defined for positive formulas only.[5]

In the special case when T and F are nonmodal, this relation turns into the consequence relation of classical propositional logic. On the other hand, there is a simple connection between this relation and (the propositional case of) the relation $\models\!\!\approx$ from [18]. The latter can be defined as follows: For any nonmodal theory T and any positive formula F, $T \models\!\!\approx F$ if, for every $I \in Mod(T)$, F is true in $(I, Mod(T))$.[6] It is clear that $T \models\!\!\approx F$ iff $\mathsf{B}T \models \mathsf{B}F$.

Even restricted to positive theories, the consequence relation $\models$ is nonmonotonic. For instance, $\mathsf{B}p \models \neg\mathsf{B}q$; this entailment will be lost as soon as $\mathsf{B}q$ is added as another axiom.

We would like to say that every axiom of a theory T is a theorem of T, but this would make sense for positive axioms only. In order to extend this assertion to arbitrary axioms, we need the following definition. The *positive form* F^+ of a formula F is the positive formula obtained from F by substituting $\neg\mathsf{B}$ for each occurence of *not*.

The positive form of any axiom of a theory T is a theorem of T. Indeed, if (I, S) is a model of T, then the axioms of T are true in (I, S, S), so that their positive forms are true in (I, S).

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 $\mathcal{L}$ (possibly with equality) by adding the modal operators B and *not*. A *theory* is now a set of sentences of the extended language; an *interpretation* is understood as in classical

[5] We do not see a reasonable definition of theoremhood for formulas containing *not*. In this respect, MBNF is similar to default logic [17], where a default can be postulated, but not derived. In logic programming, a rule can serve as a part of a program, but not as a query. In order to cover cases like these, the general definition of a "declarative formalism" in [13] distinguishes between "postulates" and "sentences."

[6] We ignore here a notational difference: Reiter's K corresponds to our B.

first-order logic. The universe of an interpretation I will be denoted by $|I|$. A *structure* is a pair (I, S), where I is an interpretation, and S is a set of interpretations with the universe $|I|$. Note that all "possible worlds" are assumed to have the same universe as the "real world" I.[7] We will call $|I|$ the *universe* of (I, S).

Let I be an interpretation, and let S^b, S^n be sets of interpretations with the universe $|I|$. We will define when a sentence is *true* in (I, S^b, 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^b, S^n) iff F is true in I.
2. $\neg F$ is true in (I, S^b, S^n) iff F is not true in (I, S^b, S^n).
3. $F \wedge G$ is true in (I, S^b, S^n) iff F and G are both true in (I, S^b, S^n).
4. $\forall x F(x)$ is true in (I, S^b, S^n) iff, for every name ξ, $F(\xi)$ is true in (I, S^b, S^n).
5. $\mathsf{B}F$ is true in (I, S^b, S^n) iff, for every $J \in S^b$, F is true in (J, S^b, S^n).
6. *not* F is true in (I, S^b, S^n) iff, for some $J \in S^n$, F is not true in (J, S^b, S^n).

A structure (I, S) is a *model* of a theory T if (i) the axioms of T are true in (I, S, S), and (ii) there is no structure (I', S') with the same universe such that S' is a proper superset of S and the axioms of T are true in (I', S', S).

If all axioms of T are nonmodal, then, as in the propositional case, there is a simple 1–1 correspondence between the models of T in the sense of MBNF and the models of T in the sense of classical logic. The models of T in the sense of MBNF are the pairs (I, S), where I is a model of T in the sense of classical logic, and S is the set of all interpretations with the universe $|I|$.

The models of $\mathsf{B}T$, where T is a nonmodal theory, are the pairs $(I, Mod_{|I|}(T))$, where $Mod_U(T)$ stands for the set of classical models of T with the universe U.

A *theorem* of a theory T is a positive sentence that is true in all models of T. As in the propositional case, the positive form of an axiom is a theorem.

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 program can be identified with the axiom set consisting of the universal closures of the formulas (7) corresponding to its rules (6), plus possibly some equality axioms. This semantics differs from the answer set semantics used in Section 5 in that it permits "non-Herbrand models."

[7] In this, we follow the approach to default logic proposed in [11].

8 Deriving New Theorems

The positive forms of the axioms of T provide an initial supply of theorems of T. Some other theorems can be obtained from this initial supply using a closure property of the set of theorems.

Closure properties of a set of positive formulas can be expressed, for instance, by saying that it is closed under a quantified modal logic defined by some collection of axioms and inference rules. We prefer another approach, based on the well-known reduction of modal operators to quantifiers.

Consider the first-order language $\mathcal{L}^\circ$, obtained from the nonmodal part $\mathcal{L}$ of the language of T as follows. We extend $\mathcal{L}$ by adding the second sort of variables $w, w', \ldots$, called *world variables*. Each function and predicate constant of $\mathcal{L}$ gets an additional world argument; in particular, each object constant turns into a unary function constant, and each propositional symbol turns into a monadic predicate. Finally, we add a new monadic predicate B, whose argument is a world variable.

For any positive formula F, its *nonmodal form* is the formula F° of the language $\mathcal{L}^\circ$, defined inductively as follows:

1. If F is an atom, then F° is obtained from F by appending w as an additional argument to each function and predicate symbol.
2. $(\neg F)^\circ$ is $\neg F^\circ$.
3. $(F \wedge G)^\circ$ is $F^\circ \wedge G^\circ$.
4. $(\forall x F)^\circ$ is $\forall x F^\circ$.
5. $(\mathsf{B}F)^\circ$ is $\forall w(B(w) \supset F^\circ)$.

For example, the nonmodal form of $\mathsf{B}P(a)$ is $\forall w(B(w) \supset P(a(w), w))$. Generally, the nonmodal form of F has w as a parameter, and we will sometimes write it as $F^\circ(w)$.

Theorem 2. *Let $F_1, \ldots, F_k, G$ be positive sentences such that the formula*

$$F_1^\circ(w) \wedge \ldots \wedge F_k^\circ(w) \supset G^\circ(w) \tag{8}$$

is universally valid. If $F_1, \ldots, F_k$ are theorems of T, then so is G.

For example, if $F_1 \wedge \ldots \wedge F_k \supset G$ is an instance of a tautology, then so is (8); consequently, the set of theorems of T is closed under propositional logic. If $\forall x P(x)$ is a theorem, then so is $P(t)$ for any ground term t, because $\forall x P(x, w) \supset P(t, w)$ is universally valid. If $\mathsf{B}(a = b)$ and $\mathsf{B}P(a)$ are theorems of T, then so is $\mathsf{B}P(b)$, because

$$\forall w(B(w) \supset a(w) = b(w)) \wedge \forall w(B(w) \supset P(a(w), w)) \supset \forall w(B(w) \supset P(b(w), w))$$

is universally valid. If $\mathsf{B}\forall x F(x)$ is a theorem, then so is $\forall x \mathsf{B} F(x)$, and vice versa, because

$$\forall w(B(w) \supset \forall x F^\circ(x, w))$$

is equivalent to

$$\forall wx(B(w) \supset F^\circ(x, w))$$

in first-order logic.

Notice that Theorem 2 is *not* applicable to F and $\mathsf{B}F$ in either direction, because neither

$$F^\circ(w) \supset \forall w(B(w) \supset F^\circ(w))$$

nor

$$\forall w(B(w) \supset F^\circ(w)) \supset F^\circ(w)$$

is universally valid. On the other hand, $\mathsf{BB}F$ is a theorem iff $\mathsf{B}F$ is, and similarly for $\mathsf{B}\neg\mathsf{B}F$ and $\neg\mathsf{B}F$.

The theorems of T that can be derived from the positive forms of its axioms by this method are "monotonic," in the sense that they will not be lost when more axioms are added to T. If, for instance, T is $\{\mathsf{B}p\}$, then $\mathsf{B}(p \vee q)$ and $\mathsf{BB}p$ are among the "monotonic" theorems of T, and $\neg\mathsf{B}q$ is not.

9 Equivalent Theories

Two theories are *equivalent* if they have the same models. Obviously, such theories have also the same theorems. Sometimes we can prove the equivalence of theories using the transformation introduced in the previous section. This transformation is extended to formulas containing negation as failure in the following way: $(not\ F)^\circ$ is $\exists w(N(w) \wedge \neg F^\circ)$, where N is a new predicate. For instance, the nonmodal form of $not\ p \supset \mathsf{B}q$ is

$$\exists w(N(w) \wedge \neg p(w)) \supset \forall w(B(w) \supset q(w)). \tag{9}$$

Theorem 3. *For any theory T and sentences F, G, if $F^\circ(w)$ is equivalent to $G^\circ(w)$ in first-order logic, then $T \cup \{F\}$ is equivalent to $T \cup \{G\}$.*

For instance, the axiom $not\ p \supset \mathsf{B}q$ can be equivalently replaced by $\mathsf{B}(not\ p \supset \mathsf{B}q)$, because (9) is equivalent to

$$\forall w[B(w) \supset (\exists w(N(w) \wedge \neg p(w)) \supset \forall w(B(w) \supset q(w)))].$$

10 Relation to Default Logic and Circumscription

Recall that a *default theory* [17] is defined by a set D of *defaults* of the form

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

and a set W of sentences that play the role of axioms. Intuitively, the default (10) says: Derive γ from α if $\beta_1, \ldots, \beta_m$ are consistent.

In accordance with the idea of Lin and Shoham [16], we identify a default (10) with (the universal closure of) the formula

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

Consistency is expressed here by the combination $\mathit{not}\neg$.

The following theorem refers to "default logic with a fixed universe"—the modification of the system of [17] proposed in [11]. The main difference is that, in the standard default logic, the parameters of an open default are treated as metavariables for ground terms, whereas the modified system handles parameters as genuine object variables.

Theorem 4. *A nonmodal sentence F is a fixed-universe consequence of a default theory (D, W) iff $D \cup \mathsf{B}W \models \mathsf{B}F$.*

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

In [11] we showed how to embed circumscription (with all nonlogical constants varied) into default logic with a fixed universe. The composition of these two transformations reduces the circumscription of P in a sentence $A(P)$ to the formula

$$\mathsf{B}A(P) \wedge \forall x(\mathit{not}\ P(x) \supset \mathsf{B}\neg P(x)). \tag{12}$$

For any nonmodal sentence F, $\mathsf{B}F$ is entailed by this formula in the sense of MBNF iff F is entailed by the circumscription in classical logic.

It is easy to check, using Theorem 3, that (12) can be equivalently written as

$$\mathsf{B}[A(P) \wedge \forall x(\mathit{not}\ P(x) \supset \neg P(x))].$$

11 Epistemic Queries

To illustrate the usefulness of epistemic queries, Reiter [18] introduces the database consisting of the following facts:

$$\begin{gathered} \mathit{Teach}(\mathit{John}, \mathit{Math}), \\ \exists x\ \mathit{Teach}(x, \mathit{CS}), \\ \mathit{Teach}(\mathit{Mary}, \mathit{Psych}) \vee \mathit{Teach}(\mathit{Sue}, \mathit{Psych}). \end{gathered} \tag{13}$$

We need to distinguish between "objective" questions, such as "which courses are taught?", and "epistemic" questions, such as "which courses are taught by known individuals?". All three courses mentioned in the database are valid answers to the first question, whereas the only answer to the second is *Math*. The first query can be represented by the formula $\exists x\, Teach(x,y)$; the second requires a modal operator: $\exists x \mathsf{B}\, Teach(x,y)$.

We will show how the approach to the semantics of epistemic queries developed by Levesque and Reiter can be reformulated in the framework of MBNF.

A *database* is a theory T (as defined in Section 7) along with a subset A of its object constants, called the *answer constants*. The role of A, as will be seen later, is to single out the symbols that are allowed to appear in answers to queries. In the example above, every object constant would be considered an answer constant. Consider, on the other hand, the modification of that example, in which the unknown teachers of computer science and psychology are represented by "null value constants":

$$\begin{gathered} Teach(John, Math), \\ Teach(Null1, CS), \\ Teach(Null2, Psych), \\ Null2 = Mary \lor Null2 = Sue. \end{gathered} \tag{14}$$

We do not want to accept the null values as answers to the queries $Teach(x, CS)$ and $Teach(x, Psych)$. Formally, this distinction can be expressed by not including *Null1* and *Null2* in the set of answer constants A.

The choice of A defines two groups of additional axioms. The *unique names axioms* are the formulas $c \neq c'$ for all pairs of distinct answer constants c, c'. The *rigid designator axioms* are the formulas $\exists x \mathsf{B}(x = c)$ for all answer constants c. These sets of axioms will be denoted, respectively, by UNA_A and RDA_A.

Let (T, A) be a database, and $F(x_1, \dots, x_k)$ a positive formula, with all parameters explicitly shown. An *answer* to $F(x_1, \dots, x_k)$ relative to (T, A) is any tuple of answer constants $(c_1, \dots, c_k)$ such that $\mathsf{B}F(c_1, \dots, c_k)$ is entailed by $\mathsf{B}T \cup UNA_A \cup RDA_A$. For closed formulas ("ground queries"), the terminology is somewhat different. The answer to F is *yes*, *no*, or *unknown*, depending on whether $\mathsf{B}T \cup UNA_A \cup RDA_A$ entails $\mathsf{B}F$, $\mathsf{B}\neg F$, or neither (assuming, of course, that it does not entail both).

For example, let T be (13) or (14), and let

$$A = \{John, Mary, Sue, Math, CS, Psych\}.$$

It is easy to verify, using Theorem 2, that *Math*, *CS* and *Psych* are answers to the query $\exists x\, Teach(x,y)$, and that *Math* is an answer to $\exists x \mathsf{B}\, Teach(x,y)$. (For the last part, the rigid designator axiom $\exists x \mathsf{B}(x = John)$ is needed.)

A theory T can be identified with the database $(T, \emptyset)$. Note that the answer to F relative to T is *yes* when $\mathsf{B}T \models \mathsf{B}F$. This condition is different from $T \models F$; it expresses that, when the axioms of T are accepted as the "explicit beliefs," F will be "implicitly believed."

Our semantics of epistemic queries, although conceptually very close to the definition from [18], is different in several ways.

1. Levesque and Reiter begin with a fixed countably infinite set of "parameters," which plays a double role. First, parameters are used to define the semantics of quantifiers, like "names" in Section 7 of this paper.[8] Second, parameters are allowed to appear in answers to database queries, like "answer constants" introduced above. In our approach, neither the set of names in the definition of a model, nor the set of answer constants in the definition of a database is assumed to be infinite or countable; moreover, the two sets are not assumed to have the same cardinality. Our semantics of quantifiers, in the case of nonmodal formulas, is identical to their classical semantics.

2. Reiter's definition is restricted to languages with parameter symbols, but without other object symbols, function symbols, or equality. For instance, it is not directly applicable to the reformulation (14) of his example, because the symbols *Null1* and *Null2* cannot be treated as parameters.

3. Reiter's databases are classical theories; our databases may include both B and *not*. This may lead to a semantics of epistemic queries for the nonmonotonic formalisms that are embeddable into MBNF, such as default logic, circumscription and logic programming. We have not investigated this in detail, but here is an example of what can be done on the basis of our definition. The axioms (13) provide no negative information about the predicate *Teach*; they are consistent with the assumption that every course is taught by every instructor. We may wish to provide some negative information by including a "closed world assumption" of some kind. The discussion of circumscription in Section 10 suggests that one way to do that may be to add the axiom

$$\forall xy(\mathit{not}\, \mathit{Teach}(x,y) \supset \neg \mathit{Teach}(x,y)).$$

Our definition of an answer to a query is applicable to nonmonotonic databases like this.

[8] Parameters are also used for this purpose in Levesque's treatment of quantified autoepistemic logic [10].

12 Conclusion

The logic of minimal belief and negation as failure provides a unified framework for several logic programming languages and nonmonotonic formalisms, as well as for the 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.

This paper does not claim that all important systems of nonmonotonic reasoning proposed in the literature can be embedded into MBNF. In particular, our system does not cover the concept of "strong introspection," introduced in [3].

Acknowledgments

I would like to thank Michael Gelfond, Benjamin Grosof, Katsumi Inoue, Hector Levesque, Fangzhen Lin, Norman McCain, Ray Reiter, Grigori Schwarz, Miroslaw Truszczyński and Thomas Woo for useful discussions on the subject of this paper.

References

[1] J. Chen. Minimal knowledge + negation as failure = only knowing (sometimes). In L. M. Pereira and A. Nerode, editors, *Logic Programming and Nonmonotonic Reasoning: Proceedings of the Second International Workshop*, pages 132–150, 1993.

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

[3] M. Gelfond. Strong introspection. In *Proc. AAAI-91*, pages 386–391, 1991.

[4] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In R. Kowalski and K. Bowen, editors, *Logic Programming: Proc. of the Fifth Int'l Conf. and Symp.*, pages 1070–1080, 1988.

[5] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.

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

[7] K. Inoue and C. Sakama. Representing abduction by positive *not*. In *ICLP-93 Postconference Workshop on Abductive Reasoning*, 1993.

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

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

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

[11] V. Lifschitz. On open defaults. In J. Lloyd, editor, *Computational Logic: Symposium Proceedings*, pages 80–95. Springer, 1990.

[12] V. Lifschitz. Nonmonotonic databases and epistemic queries. In *Proc. of IJCAI-91*, pages 381–386, 1991.

[13] V. Lifschitz. Restricted monotonicity. In *Proc. AAAI-93*, pages 432–437, 1993.

[14] V. Lifschitz and T. Woo. Answer sets in general nonmonotonic reasoning (preliminary report). In B. Nebel, C. Rich, and W. Swartout, editors, *Proc. of the Third Int'l Conf. on Principles of Knowledge Representation and Reasoning*, pages 603–614, 1992.

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

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

[17] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1,2):81–132, 1980.

[18] R. Reiter. What should a database know? *Journal of Logic Programming*, 14:127–153, 1992.

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

Appendix. Proofs of Theorems

In order to prove Theorem 1 (Section 5), we need to establish a few lemmas. By *Lit* we denote the set of all (propositional) literals. A set of literals is *closed* if it is consistent or equals *Lit*. For further reference, we collected in the following lemma some simple properties of $Mod(M)$ for closed sets of literals M.

Lemma 1. *Let M be a closed set of literals, M' a set of literals, L a literal, I an interpretation, S a set of interpretations.*

a. *$Mod(M) \subset Mod(M')$ iff $M' \subset M$.*

b. *$\mathsf{B}L$ is true in $(I, Mod(M), S)$ iff $L \in M$.*

c. *not L is true in $(I, S, Mod(M))$ iff $L \notin M$.*

We say that two sets of interpretations S_1, S_2 are *equivalent* if

$$\bigcap_{I \in S_1} I = \bigcap_{I \in S_2} I, \quad \bigcup_{I \in S_1} I = \bigcup_{I \in S_2} I.$$

It is clear that if S_1 and S_2 are equivalent then any formula of the form $\mathsf{B}L$, where L is a literal, is true in (I, S_1, S) iff it is true in (I, S_2, S). Consequently, the same can be said about any formula of the form (7).

Lemma 2. *For any set of interpretations S, there exists a closed set of literals M such that $Mod(M)$ contains S and is equivalent to it.*

Proof. Take

$$S' = \left\{ J \; : \; \bigcap_{I \in S} I \subset J \subset \bigcup_{I \in S} I \right\}.$$

It is clear that S' contains S and is equivalent to it. On the other hand, $S' = Mod(M)$, where M obtained from $\bigcap_{I \in S} I$ by adding the negations of all atoms that do not belong to $\bigcup_{I \in S} I$. If $S = \emptyset$ then $M = Lit$; otherwise, M does not contain complementary literals. Consequently, M is closed.

The definition of a model for propositional MBNF (Section 4) can be reformulated using the following notation. For any two structures (I, S) and (I', S'), we will write $(I, S) < (I', S')$ if S is a proper subset of S'. For any theory T and any set of interpretations S, let $\Gamma(T, S)$ be the set of all structures (I, S'), maximal with respect to the relation $<$, such that the axioms of T are true in (I, S', S). Then a structure (I, S) is a model of T iff $(I, S) \in \Gamma(T, S)$.

Using Lemma 2, we can characterize the models of a disjunctive program in terms of a simplified version of the operator Γ. For any disjunctive program Π and any set $M \subset Lit$, let $\Gamma_0(\Pi, M)$ be the set of all minimal closed sets $M' \subset Lit$ which satisfy the condition:

$$\text{The rules of } \Pi \text{ are true in } (I, Mod(M'), Mod(M)). \tag{15}$$

Here I is an arbitrary interpretation; it is easy to see that the choice of I has no effect on whether or not a rule is true relative to a triple (I, S^b, S^n).

Lemma 3. *A structure (I, S) is a model of a disjunctive program Π iff S has the form $Mod(M)$ for a set of literals M such that $M \in \Gamma_0(\Pi, M)$.*

Proof. Let (I, S) be a model of Π, so that $(I, S) \in \Gamma(\Pi, S)$. Recall that $\Gamma(\Pi, S)$ is the set of all maximal structures (I, S') satisfying the condition:

$$\text{The rules of } \Pi \text{ are true in } (I, S', S). \tag{16}$$

In particular,

$$\text{The rules of } \Pi \text{ are true in } (I, S, S). \tag{17}$$

Take a closed set of literals M such that $Mod(M)$ contains S and is equivalent to it (Lemma 2). We want to show that $S = Mod(M)$. Assume that this is not the case, so that S is a proper subset of $Mod(M)$. Then $(I, S) < (I, Mod(M))$. Since $Mod(M)$ is equivalent to S, the rules of Π are true in $(I, Mod(M), S)$. This means that $(I, Mod(M))$ is among the structures (I, S') satisfying (16), which contradicts the maximality of (I, S). This contradiction shows that $S = Mod(M)$.

We conclude now from (17) that the rules of Π are true in $(I, Mod(M), Mod(M))$, so that M satisfies (15) as M'. In order to show that $M \in \Gamma_0(\Pi, M)$, we need only to check that no proper closed subset M' of M satisfies (15). Assume that M' is such a subset. Since $S = Mod(M)$, this means that the rules of Π are true in $(I, Mod(M'), S)$, so that $(I, Mod(M'))$ is among the structures (I, S') satisfying (16). By Lemma 1, Part *a*, $Mod(M')$ is a proper superset of $Mod(M)$, that is, of S, which contradicts the maximality of (I, S).

Now assume that $M \in \Gamma_0(\Pi, M)$. Take any interpretation I. We need to prove that $(I, Mod(M))$ is a model of Π, which means that $(I, Mod(M)) \in \Gamma(\Pi, Mod(M))$. It follows from the definition of Γ_0 that M is closed, and the rules of Π are true in $(I, Mod(M), Mod(M))$. In other words, $Mod(M)$ satisfies the condition

$$\text{The rules of } \Pi \text{ are true in } (I, S', Mod(M)) \tag{18}$$

as S'. By the definition of Γ, it remains to show that this condition is not satisfied for any proper superset S' of $Mod(M)$. Assume that S' is a superset of $Mod(M)$ satisfying (18). By Lemma 2, there is a closed set of literals M' such that $Mod(M')$

contains S' and is equivalent to it. The rules of Π are true in $(I, Mod(M'), Mod(M))$, so that M' satisfies (15). But $Mod(M) \subset S' \subset Mod(M')$, so that, by Lemma 1, Part a, M' is a subset of M. Since $M \in \Gamma_0(\Pi, M)$, it follows that $M = M'$.

Lemma 3 is proved.

The notion of an answer set [5] is defined in two steps. First let Π be a disjunctive program that doesn't contain *not* (in every rule (6), $m = n$). Then the answer sets of Π are the minimal closed subsets M of *Lit* that satisfy the condition: For each rule

$$L\Delta 1 \mid \ldots \mid L\Delta l \,\text{¸}\, L\Delta l+1, \ldots, L\Delta m$$

from Π, if $L\Delta l+1, \ldots, L\Delta m \Sigma M$, then, for some $i = 1, \ldots, l$, $L\Delta i \Sigma M$. Now let Π be any disjunctive program. For any set $M \imath Lit$, let Π^M be the disjunctive program obtained from Π by deleting

(i) each rule that has a formula *not* L in its body with $L \Sigma M$, and

(ii) all formulas of the form *not* L in the bodies of the remaining rules.

Clearly, Π^M doesn't contain *not*, so that the answer sets of Π^M are already defined. If M is among them, we say that M is an answer set of Π.

Lemma 4. *For any disjunctive program* Π *which doesn't contain not, and any set of literals* M*,* $\Gamma_0(\Pi, M)$ *is the set of all answer sets of* Π.

Proof. If Π which doesn't contain *not*, then each of its rules has the form

$$\mathsf{B}L_{l+1} \wedge \ldots \wedge \mathsf{B}L_m \supset \mathsf{B}L_1 \vee \ldots \vee \mathsf{B}L_l. \tag{19}$$

Lemma 1, Part b, shows that the definition of Γ_0 can be stated in this case as follows: $\Gamma_0(\Pi, M)$ is the set of all minimal closed sets $M' \subset Lit$ such that, for every rule (19) from Π, if $L_{l+1}, \ldots, L_m \in M'$, then, for some $i = 1, \ldots, l$, $L_i \in M'$. This condition characterizes the answer sets of Π.

Lemma 5. *For any disjunctive program* Π *and any closed set of literals* M*,*

$$\Gamma_0(\Pi^M, M) = \Gamma_0(\Pi, M).$$

Proof. By the definition of Γ_0, it is sufficient to check that, for any interpretation I and any closed sets of literals M, M', the following two conditions are equivalent:

$$\text{All rules of } \Pi \text{ are true in } (I, Mod(M'), Mod(M)), \tag{20}$$

$$\text{All rules of } \Pi^M \text{ are true in } (I, Mod(M'), Mod(M)). \tag{21}$$

Assume (20). Any rule of Π^M is obtained from one of the rules (7) of Π such that $L_{m+1}, \ldots, L_n \notin M$ by dropping the subformulas *not* L_{m+1},...,*not* L_n; by Lemma 1, Part *c*, all these subformulas are true in $(I, Mod(M'), Mod(M))$. Assume now (21), and take any rule (7) of Π. If, for some $i = m+1, \ldots, n$, $L_i \in M$, then, by Lemma 1, Part *c*, the formula $\neg L_i$ is true in $(I, Mod(M'), Mod(M))$, and, consequently, so is (7). Otherwise, Π_M contains a rule which is obtained from (7) by dropping some conjunctive members in the antecedent, and which consequently implies (7) in propositional logic. Lemma 5 is proved.

Now we are ready to prove Theorem 1. It is sufficient to prove the last part, because the first two parts are its special cases.

Theorem 1, Part C. *The models of a disjunctive program* Π *are the structures of the form* $(I, Mod(M))$*, where* M *is an answer set of* Π.

Proof. By Lemma 3, it is sufficient to show that, for any set of literals M, M is an answer set of Π iff $M \in \Gamma_0(\Pi, M)$. By the definition of answers sets, the first condition means that M is the answer set of Π^M. By Lemma 4, this can be expressed by the formula $M \in \Gamma_0(\Pi^M, M)$. By Lemma 5, this is equivalent to $M \in \Gamma_0(\Pi, M)$.

The proof of Theorem 2 is based on the following construction. For any structure (I, S), we define the interpretation I° for the language $\mathcal{L}^\circ$ as follows. The universe of the world variables is the set of all interpretations J of $\mathcal{L}$ such that $|I| = |J|$. The function $I^\circ[\![f]\!]$, representing a function constant f in I°, is defined by

$$I^\circ[\![f]\!](\xi, J) = J[\![f]\!](\xi).$$

The set $I^\circ[\![P]\!]$, representing a predicate constant P in I°, is defined by

$$(\xi, J) \in I^\circ[\![P]\!] \text{ iff } \xi \in J[\![P]\!].$$

Finally, B is represented by S.

Lemma 6. *A positive sentence* F *is true in* (I, S) *iff* I *satisfies* $F^\circ(w)$ *in* I°.

This assertion, extended to the sentences that may include names (Section 7), is easily verified by induction on F.

Theorem 2. *Let* F *and* G *be positive sentences such that (8) is universally valid. If* F *is a theorem of* T*, then so is* G.

Proof. Let (I, S) be a model of T. Consider the interpretation I° defined as above. Since $F_1, \ldots, F_k$ are theorems of T, they are true in (I, S). Then, by Lemma 6, I

satisfies $F_1^\circ(w)$,...,$F_k^\circ(w)$ in I°. Since (8) is universally valid, I also satisfies $G^\circ(w)$ in I°. By Lemma 6, it follows that G is true in (I, S).

In order to prove Theorem 3, we need to generalize the definition of I° as follows. We begin with a triple (I, S^b, S^n), as in Section 4, and define I° as above, except that B is represented by S^b, and the new predicate N is represented by S^n. Now Lemma 6 can be generalized as follows:

Lemma 7. *A sentence F is true in (I, S^b, S^n) iff I satisfies $F^\circ(w)$ in I°.*

Theorem 3. *For any theory T and sentences F, G, if $F^\circ(w)$ is equivalent to $G^\circ(w)$ in first-order logic, then $T \cup \{F\}$ is equivalent to $T \cup \{G\}$.*

Proof: It follows from Lemma 7 that the axioms of $T \cup \{F\}$ are true in a triple (I, S^b, S^n) iff the axioms of $T \cup \{G\}$ are true in it.

Let us turn now to Theorem 4 (Section 10). The proof uses the following generalization of the operator Γ to MBNF with quantifiers. For any theory T, nonempty set U, and set S of interpretations with the universe U, by $\Gamma_U(T, S)$ we denote the set of all maximal structures (I, S') with the universe U such that the axioms of T are true in (I, S', S). Then a structure (I, S) is a model of T iff $(I, S) \in \Gamma_{|I|}(T, S)$.

The notion of a fixed-universe consequence of a default theory (D, W) is defined as follows [11]. Let U be a nonempty set. By $\mathcal{L}$ we denote the language of (D, W); $\mathcal{L}^*$ is the language obtained from $\mathcal{L}$ by adding names for the elements of U. For any set S of interpretations with the universe U, let $Th(S)$ $[Th^*(S)]$ be the set of sentences of $\mathcal{L}$ [respectively, $\mathcal{L}^*$] that are true in all structures from S.

For any set S of interpretations with the universe U, consider the largest set $S' \subset Mod_U(W)$ which satisfies the condition: For any default

$$\alpha(x) : \beta_1(x), \ldots, \beta_m(x) / \gamma(x) \tag{22}$$

from D (with all parameters explicitly shown) and any tuple of names ξ,

$$\text{if } \alpha(\xi) \in Th^*(S') \text{ and } \neg\beta_1(\xi), \ldots, \neg\beta_m(\xi) \notin Th^*(S) \text{ then } \gamma(\xi) \in Th^*(S'). \tag{23}$$

This largest S' always exists; we denote it by $\Delta(S)$.

A sentence F is a *fixed-universe consequence* of (D, W) if, for every nonempty U and every fixpoint S of Δ_U, F is true in all structures from S.

The following lemma relates Δ_U to Γ_U.

Lemma 8. *For any default theory (D, W), any nonempty set U, and any set S of interpretations with the universe U, $\Gamma_U(D \cup \mathsf{B}W, S)$ is the set of all structures of the form $(I, \Delta_U(S))$.*

Proof. It is sufficient to check that, for any interpretation I and sets S, S' of interpretations with the universe $|I|$, the following conditions are equivalent:

(i) all formulas from $D \cup \mathsf{B}W$ are true in (I, S', S),

(ii) $S' \subset \mathit{Mod}_{|I|}(W)$ and, for every default (22) from D, S' satisfies (23).

To say that all formulas from $\mathsf{B}W$ are true in (I, S', S) means to say that, for each $F \in W$, F is true in every interpretation from S', that is, $S' \subset \mathit{Mod}_{|I|}(W)$. On the other hand, the formula

$$\forall x[\mathsf{B}\alpha(x) \wedge \mathit{not}\neg\beta_1(x) \wedge \ldots \wedge \mathit{not}\neg\beta_m(x) \supset \mathsf{B}\gamma(x)],$$

representing the default (22), is true in (I, S', S) iff the condition (23) holds.

Lemma 9. *The models of $D \cup \mathsf{B}W$ are the pairs (I, S) such that S is a fixpoint of $\Delta_{|I|}$.*

Proof: Immediately follows from Lemma 8.

Theorem 4. *A nonmodal sentence F is a fixed-universe consequence of a default theory (D, W) iff $D \cup \mathsf{B}W \models \mathsf{B}F$.*

Proof: Immediately follows from Lemma 9.