Extended Logic Programs as Autoepistemic Theories

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

Grigori Schwarz
Robotics Laboratory
Computer Science Department
Stanford University
Stanford, CA 94305, USA

Abstract

Recent research on applications of nonmonotonic reasoning to the semantics of logic programs demonstrates that some nonmonotonic formalisms are better suited for such use than others. Circumscription is applicable as long as the programs under consideration are stratified. To describe the semantics of general logic programs without the stratification assumption, one has to use autoepistemic logic or default logic. When Gelfond and Lifschitz extended this work to programs with classical negation, they used default logic, because it was not clear whether autoepistemic logic could be applied in that wider domain. In this paper we show that programs with classical negation can be, in fact, easily represented by autoepistemic theories. We also prove that an even simpler embedding is possible if reflexive autoepistemic logic is used. Both translations are applicable to disjunctive programs as well.

1 Introduction

Recent research on applications of nonmonotonic reasoning to the semantics of logic programs demonstrates that some nonmonotonic formalisms are better suited for such use than others. *Circumscription* is applicable as long as the programs under consideration are stratified [10]. To describe the semantics of general logic programs without the stratification assumption, one has to use *autoepistemic logic* [4], [5] or *default logic* [1], [2]. When Gelfond and Lifschitz extended this work to programs with classical negation, they used default logic, because it was not clear whether autoepistemic logic could be applied in that wider domain.

In this paper we show that programs with classical negation can be, in

fact, easily represented by autoepistemic theories. The new translation is applicable to disjunctive programs as well. This last fact is particularly striking, because disjunctive rules do not seem to be reducible to defaults [7].

Recall that a *general logic program* is a set of rules of the form

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

where each A_i is an atom. Gelfond's transformation maps such a rule into the axiom

$$A_1 \wedge \ldots \wedge A_m \wedge \neg \mathrm{B} A_{m+1} \wedge \ldots \wedge \neg \mathrm{B} A_n \supset A_0, \tag{2}$$

where B is the "belief" operator of autoepistemic logic.[1] The declarative semantics of a program can be characterized in terms of the autoepistemic theory obtained by this transformation ([4], Theorem 5; [5], Theorem 3).

An *extended logic program* consists of rules of the same form (1), except that each A_i is allowed to be a literal (an atom possibly preceded by $\neg$). Thus an extended rule may contain two kinds of negation—classical negation $\neg$ and negation as failure *not*. Such rules are useful for representing incomplete information. Their semantics, defined in terms of "answer sets" [6], is noncontrapositive, in the sense that it distinguishes between the rules $P \leftarrow Q$ and $\neg Q \leftarrow \neg P$. The former is, intuitively, an "inference rule" allowing us to derive P from Q; the latter allows us to derive $\neg Q$ from $\neg P$. For example, the answer set of the program

$$\begin{array}{l} Q \leftarrow \\ P \leftarrow Q \end{array}$$

is $\{P, Q\}$; the answer set of

$$\begin{array}{l} Q \leftarrow \\ \neg Q \leftarrow \neg P \end{array}$$

is $\{Q\}$.

When applied to an extended rule, Gelfond's transformation may distort its meaning. For instance, it maps $P \leftarrow Q$ and $\neg Q \leftarrow \neg P$ into equivalent formulas, $Q \supset P$ and $\neg P \supset \neg Q$.

Can we come up with a "noncontrapositive" modification of Gelfond's mapping? One possibility could be to insert B before each literal in the rule, not only before the literals preceded by *not*, so that (1) will be represented by

$$\mathrm{B} A_1 \wedge \ldots \wedge \mathrm{B} A_m \wedge \neg \mathrm{B} A_{m+1} \wedge \ldots \wedge \neg \mathrm{B} A_n \supset \mathrm{B} A_0. \tag{3}$$

This transformation maps $P \leftarrow Q$ and $\neg Q \leftarrow \neg P$ into nonequivalent axioms, $\mathrm{B}Q \supset \mathrm{B}P$ and $\mathrm{B}\neg P \supset \mathrm{B}\neg Q$. However, this idea does not work: The program consisting of just one rule with the empty body, $P \leftarrow$, would correspond to the autoepistemic theory $\{\mathrm{B}P\}$, which has no stable expansions.[2]

Considerations of this sort have led the authors of [6] to the rejection of autoepistemic logic as an instrument for the study of logic programming. We prove, however, in this paper that a simple hybrid of (2) and (3) does the job. We propose to replace every literal A_i in (1) that is not preceded by the operator *not* by the conjunction $A_i \wedge \mathrm{B}A_i$ ("A_i is a true belief"), so that (1) will be represented by the axiom

$$(A_1 \wedge \mathrm{B}A_1) \wedge \ldots \wedge (A_m \wedge \mathrm{B}A_m) \wedge \neg\mathrm{B}A_{m+1} \wedge \ldots \wedge \neg\mathrm{B}A_n \supset (A_0 \wedge \mathrm{B}A_0). \quad (4)$$

For instance, the rule $P \leftarrow$ turns into the axiom $P \wedge \mathrm{B}P$.

We show that this transformation correctly represents the meaning of a rule, in the sense that there is a one-to-one correspondence between the consistent answer sets of an extended program and the consistent stable expansions of the autoepistemic theory whose axioms are obtained in this way from its rules. Specifically, the propositional closure of each answer set is the nonmodal part of the corresponding stable expansion.

The same result holds for disjunctive programs, if a disjunctive rule

$$A_1 \mid \ldots \mid A_l \leftarrow A_{l+1}, \ldots, A_m, \mathit{not}\ A_{m+1}, \ldots, \mathit{not}\ A_n \quad (5)$$

is transformed into the autoepistemic axiom

$$\begin{aligned}(A_{l+1} \wedge \mathrm{B}A_{l+1}) \wedge \ldots \wedge (A_m \wedge \mathrm{B}A_m) \wedge \neg\mathrm{B}A_{m+1} \wedge \ldots \wedge \neg\mathrm{B}A_n \\ \supset (A_1 \wedge \mathrm{B}A_1) \vee \ldots \vee (A_l \wedge \mathrm{B}A_l).\end{aligned} \quad (6)$$

Logic programs can be also translated into *reflexive autoepistemic logic*—the modification of autoepistemic logic introduced in [19]. That translation is even simpler; the axiom corresponding to (5) is, in this case,

$$\mathrm{B}A_{l+1} \wedge \ldots \wedge \mathrm{B}A_m \wedge \mathrm{B}\neg\mathrm{B}A_{m+1} \wedge \ldots \wedge \mathrm{B}\neg\mathrm{B}A_n \supset \mathrm{B}A_1 \vee \ldots \vee \mathrm{B}A_l. \quad (7)$$

Our results are, in fact, slightly more general. They are stated in terms of propositional combinations of "protected literals" of the logic of minimal belief and negation as failure (MBNF) [12], which include disjunctive programs as a special case.

The correspondence between logic programs and reflexive autoepistemic theories, given by translation (7), was independently found by Marek and Truszczyński [16]. They also analyze translations (6) and (7) in detail, and stress the special role of reflexive autoepistemic logic for the analysis of the semantics of extended logic programs.

The correspondence between logic programs and autoepistemic logic, given by translation (6), was independently found by Jianhua Chen [3]. Interestingly, he also uses logic MBNF as a starting point of his considerations.

In Section 2, we give a brief review of three modal nonmonotonic systems: autoepistemic logic, reflexive autoepistemic logic, and the propositional fragment of the logic of minimal belief and negation as failure. The main results are stated in Section 3 and proved in Section 4.

2 Modal Nonmonotonic Logics

Formulas of autoepistemic logic are built from propositional atoms using propositional connectives and the modal operator B. Formulas of MBNF may contain, in addition, a second modal operator, *not*. We will distinguish between the two languages by calling their formulas *unimodal* and *bimodal*, respectively. Formulas not containing modal operators will be called *nonmodal*.

An *interpretation* is a set of atoms. A *unimodal structure* is a pair (I, S), where I is an interpretation, and S a set of interpretations. A *bimodal structure* is a triple (I, S^b, S^n), where I is an interpretation, and S^b, S^n are sets of interpretations.

2.1 Autoepistemic Logic

For any sets T, E of unimodal formulas, E is said to be a *stable expansion* of T if it satisfies the equation

$$E = \{\psi : T \cup \{\neg \mathrm{B}\varphi : \varphi \notin E\} \cup \{\mathrm{B}\varphi : \varphi \in E\} \vdash \psi\}$$

[18]. Intuitively, T is a "theory," the elements of T are its "axioms," and the elements of E are the "theorems" that follow from the axioms in autoepistemic logic.

Autoepistemic logic can be also described in terms of models [17]. The satisfaction relation $\models_{ae}$ between a unimodal structure and a unimodal formula is defined inductively, as follows. For an atom φ, $(I, S) \models_{ae} \varphi$ iff $\varphi \in I$. For any formula φ, $(I, S) \models_{ae} \mathrm{B}\varphi$ iff, for every $J \in S$, $(J, S) \models_{ae} \varphi$. The propositional connectives are handled in the usual way. Now we can define the notion of a model: For a set T of unimodal formulas and a set S of interpretations, S is said to be an *autoepistemic model* of T if it satisfies the equation

$$S = \{I : \text{for each } \varphi \in T,\ (I, S) \models_{ae} \varphi\}.$$

For any set S of interpretations, by $Th(S)$ we denote the *theory* of S—the set of all formulas φ such that, for every $I \in S$, $(I, S) \models_{ae} \varphi$.

The relationship between stable expansions and autoepistemic models is described by the following proposition:

Proposition 2.1 *For any sets T, E of unimodal formulas, E is a consistent stable expansion of T if and only if $E = Th(S)$ for some nonempty autoepistemic model S of T.*

This fact may be extracted from [17]. It is also presented in [8], in somewhat different terms, and is discussed in [20] in more detail.

2.2 Reflexive Autoepistemic Logic

For any sets T, E of unimodal formulas, E is said to be a *reflexive expansion* of T if it satisfies the equation

$$E = \{\psi : T \cup \{\neg \mathrm{B}\varphi : \varphi \notin E\} \cup \{\varphi \equiv \mathrm{B}\varphi : \varphi \in E\} \vdash \psi\}$$

[19], [14]. Note the difference between this definition and Moore's definition of a stable expansion: Positive introspection for a formula $\varphi \in E$ is represented by the term $\varphi \equiv \mathrm{B}\varphi$, rather than $\mathrm{B}\varphi$.

Reflexive expansions admit a semantical characterization similar to the one given above for stable expansions. The definition of the satisfaction relation $\models_{rae}$ is similar to the definition of $\models_{ae}$, except that the clause for B reads as follows: $(I,S) \models_{rae} \mathrm{B}\varphi$ iff, for every $J \in \{I\} \cup S$, $(J,S) \models_{rae} \varphi$. We say that S is a *reflexive autoepistemic model* of T if

$$S = \{I : \text{for each } \varphi \in T,\ (I,S) \models_{rae} \varphi\}.$$

Clearly, if $I \in S$ then the conditions $(I,S) \models_{rae} \varphi$ and $(I,S) \models_{ae} \varphi$ are equivalent. It follows that $Th(S)$ can be equivalently described as the set of all formulas φ such that, for every $I \in S$, $(I,S) \models_{rae} \varphi$.

The following counterpart of Proposition 2.1 is proved in [20].

Proposition 2.2 *For any sets T, E of unimodal formulas, E is a consistent reflexive expansion of T if and only if $E = Th(S)$ for some nonempty reflexive autoepistemic model S of T.*

There exist simple translations from reflexive autoepistemic logic into autoepistemic logic and back [19]. We will need the following fact, which easily follows from the definitions:

Proposition 2.3 *For any nonmodal formula φ and any unimodal structure (I,S),*

(a) $(I,S) \models_{rae} \mathrm{B}\varphi$ if and only if $(I,S) \models_{ae} \varphi \wedge \mathrm{B}\varphi$,

(b) If $S \neq \emptyset$ then $(I,S) \models_{ae} \mathrm{B}\varphi$ if and only if $(I,S) \models_{rae} \neg\mathrm{B}\neg\mathrm{B}\varphi$.

2.3 The Logic of Minimal Belief and Negation as Failure

MBNF, the logic of minimal belief and negation as failure, is defined in [11][3]. Here we only consider its propositional fragment.

The satisfaction relation $\models_{mbnf}$ between a bimodal structure and a bimodal formula is defined inductively, with the usual clauses for atoms and propositional connectives, and the following clauses for the modal operators: $(I,S^b,S^n) \models_{mbnf} \mathrm{B}\varphi$ iff, for every $J \in S^b$, $(J,S^b,S^n) \models_{mbnf} \varphi$; $(I,S^b,S^n) \models_{mbnf} \mathit{not}\ \varphi$ iff, for some $J \in S^n$, $(J,S^b,S^n) \not\models_{mbnf} \varphi$.

Let T be a set of bimodal formulas. We write $(I, S^b, S^n) \models_{mbnf} T$ if $(I, S^b, S^n) \models_{mbnf} \varphi$ for each $\varphi \in T$. A unimodal structure (I, S) is an *MBNF-model* of T if $(I, S, S) \models_{mbnf} T$ and, for every proper superset S' of S, $(I, S', S) \not\models_{mbnf} T$.

In this paper, we mostly deal with *modalized* formulas, that is, formulas in which every occurrence of an atom is in the scope of a modal operator. It is easy to see that, for modalized φ, the relation $(I, S^b, S^n) \models_{mbnf} \varphi$ does not depend on I. Consequently, if all formulas in T are modalized, then the relation "(I, S) is an MBNF-model of T" does not depend on I.

Protected literals are formulas of the forms $\mathrm{B}\varphi$ and *not* φ, where φ is a literal. If every formula in T is a propositional combination of protected literals, then the models of T have a particularly simple structure: Each of them has the form $(I, Mod(M))$, where M is a set of literals. (For any set M of nonmodal formulas, $Mod(M)$ stands for the set of models of M in the sense of propositional logic—the set of all interpretations that make the formulas from M true.) Moreover, one can define, for any such T, when a set of literals is an "answer set" of T, so that the models of T can be characterized as the pairs $(I, Mod(M))$ for all answer sets M of T [12]. For our purposes, the exact definition of this concept is inessential. We only need to know that it is a generalization of the definition of an answer set for disjunctive logic programs [6], provided that we agree to identify a rule (5) with the bimodal formula

$$\mathrm{B}A_{l+1} \wedge \ldots \wedge \mathrm{B}A_m \wedge \mathit{not}\ A_{m+1} \wedge \ldots \wedge \mathit{not}\ A_n \supset \mathrm{B}A_1 \vee \ldots \vee \mathrm{B}A_l.^4 \quad (8)$$

The property of answer sets mentioned above ([12], Theorem 1) can be stated as follows:

Proposition 2.4 *Let T be a set of propositional combinations of protected literals. A unimodal structure (I, S) is an MBNF-model of T if and only if $S = Mod(M)$ for some answer set M of T.*

3 Main Results

Let φ be a propositional combination of protected literals. Define φ^a and φ^r to be the unimodal formulas obtained from φ as follows:

- φ^a is the result of replacing each protected literal $\mathrm{B}\psi$ by $\psi \wedge \mathrm{B}\psi$, and each protected literal *not* ψ by $\neg\mathrm{B}\psi$;
- φ^r is the result of replacing each protected literal *not* ψ by $\mathrm{B}\neg\mathrm{B}\psi$.

Furthermore, if T is a set of propositional combinations of protected literals, we define:

$$T^a = \{\varphi^a : \varphi \in T\}, \quad T^r = \{\varphi^r : \varphi \in T\}.$$

It is clear that if φ has the form (8), then φ^a is (6), and φ^r is (7). Consequently, when applied to logic programs, the mappings $T \mapsto T^a$ and

$T \mapsto T^r$ turn into the two representations of programs by formulas discussed in the introduction.

The following theorem shows that these mappings correctly represent the semantics of bimodal formulas in autoepistemic logic and reflexive autoepistemic logic, respectively.

Main Theorem. *Let T be a set of propositional combinations of protected literals. For any interpretation I and any nonempty set of interpretations S, the following conditions are equivalent:*

(i) (I, S) is an MBNF-model of T,

(ii) S is an autoepistemic model of T^a,

(iii) S is a reflexive autoepistemic model of T^r.

Moreover, for any consistent set M of literals, the following conditions are equivalent:

(iv) M is an answer set of T,

(v) $Th(Mod(M))$ is a stable expansion of T^a,

(vi) $Th(Mod(M))$ is a reflexive expansion of T^r.

Moreover, each consistent stable (reflexive) expansion of T^a (of T^r) has the form $Th(Mod(M))$ for some consistent set M of literals.

Without the assumption that S is nonempty (or M consistent), the assertions of the theorem would be incorrect. Take, for instance, T to be any of the sets $\{not\ p\}$, $\{\neg \mathrm{B}p, \neg \mathrm{B}\neg p\}$, or

$$\{\neg \mathrm{B}p \vee \mathrm{B}\neg p, \mathrm{B}p \vee \neg \mathrm{B}\neg p\}.$$

(The last example can be written as the program $\{\neg p \leftarrow p, p \leftarrow \neg p\}$.) In each case, $\emptyset$ is an autoepistemic model of T^a and a reflexive autoepistemic model of T^r, and it does not correspond to any MBNF-model of T.

As an immediate corollary, we get an autoepistemic interpretation of disjunctive logic programs with classical negation. An *extended disjunctive program* is a set Π of rules of the form (5), where each A_i is a literal. By Π^a we will denote the modal theory obtained from Π by replacing each rule (5) with the modal formula (6). By Π^r we denote the modal theory obtained by replacing each rule of the form (5) with the formula (7).

Corollary 3.1 *Let Π be an extended disjunctive program. For any consistent set M of literals, the following conditions are equivalent:*

(i) M is an answer set of Π,

(ii) $Th(Mod(M))$ is a stable expansion of Π^a,

(iii) $Th(Mod(M))$ *is a reflexive expansion of* Π^r.

Moreover, each consistent stable (reflexive) expansion of Π^a *(of* Π^r*) has the form* $Th(Mod(M))$ *for some consistent set* M *of literals.*

Corollary 3.1 applies, in particular, to general logic programs, when Gelfond's translation, transforming (1) into (2) [4], is applicable also. In this special case, there is an essential difference between Gelfond's translation G and our translation $\Pi \mapsto \Pi^a$. The main property of G is that there is a one-to-one correspondence between the answer sets of Π and the stable expansions of $G(\Pi)$, such that an answer set coincides with the set of atoms of the corresponding stable expansion. Because different stable sets can have the same atoms, it may happen that two programs have the same answer sets, but their G-translations have different stable expansions. In other words, G can transform two equivalent logic programs into nonequivalent autoepistemic theories. For the translation $\Pi \mapsto \Pi^a$, the stable expansion corresponding to an answer set M equals $Th(Mod(M))$, so that it is uniquely determined by M.

Consider, for example, two logic programs: $\Pi_1 = \{p \leftarrow q\}$ and $\Pi_2 = \{p \leftarrow p\}$. The only answer set of each program is $\emptyset$. However, the stable expansions of their G-translations are different. Indeed,

$$G(\Pi_1) = \{q \supset p\},\ G(\Pi_2) = \{p \supset p\};$$

the only stable expansion of $G(\Pi_1)$ is $Th(Mod\{q \supset p\})$, and the only stable expansion of $G(\Pi_2)$ is $Th(Mod(\emptyset))$. For our translation,

$$\Pi_1^a = \{(Bq \wedge q) \supset (Bp \wedge p)\},\ \Pi_2^a = \{(Bp \wedge p) \supset (Bp \wedge p)\};$$

each theory has $Th(Mod(\emptyset))$ as the only stable expansion.

The proof of the main theorem is based on the "main lemma" stated below. In the statement of the lemma, every axiom F is required to satisfy the following condition: Each occurrence of an atom in F is a part of a protected literal. Such formulas are called *formulas with protected literals*, or *PL-formulas* [12]. Alternatively, PL-formulas can be characterized as the formulas built from protected literals using propositional connectives and the operators B and *not*. Obviously, this includes propositional combinations of protected literals as a special case.

We say that a unimodal structure (I, S) *locally models* a set T of bimodal formulas if $(I, S, S) \models_{mbnf} T$ and, for every interpretation $J \notin S$,

$$(I, S \cup \{J\}, S) \not\models_{mbnf} T.$$

This definition is similar to the definition of an MBNF-model (Section 2.3), except that, instead of arbitrary supersets of S, we consider the supersets obtained from S by adding exactly one interpretation J.[5] It is clear that every MBNF-model of T locally models T. The main lemma asserts that the converse also holds if every axiom of T is a PL-formula:

Main Lemma. *Let T be a set of PL-formulas, and let (I, S) be a unimodal structure with $S \neq \emptyset$. If (I, S) locally models T, then it is an MBNF-model of T.*

4 Proofs

Proof of the Main Lemma. Let T be a set of PL-formulas, and let (I, S) be a structure which locally models T, with $S \neq \emptyset$. Assume that (I, S) is not an MBNF-model of T. Then, for some proper superset S' of S, $(I, S', S) \models T$.

Let $G \in S' \setminus S$. Define the interpretation J as follows: For any atom p,

(a) if $p \in H$ for each $H \in S$, then $p \in J$ iff $p \in H$ for each $H \in S'$;

(b) if $p \notin H$ for each $H \in S$, then $p \in J$ iff $p \in H$ for some $H \in S'$;

(c) if none of the above holds, then $p \in J$ iff $p \in G$.

(The conditions in (a) and (b) cannot apply simultaneously, because S is nonempty.)

First we will show that $J \notin S$. Assume that $J \in S$. Since $G \notin S$, it follows that $J \neq G$. Take an atom p which belongs to one of the sets J, G, but not to the other. It is clear that case (c) from the definition of J does not apply to p. If case (a) applies, that is, $p \in H$ for each $H \in S$, then, in particular, $p \in J$. Consequently, $p \in H$ for each $H \in S'$, and, in particular, $p \in G$, which contradicts the choice of p. If case (b) applies, that is, $p \notin H$ for each $H \in S$, then, in particular, $p \notin J$. Consequently, $p \notin H$ for each $H \in S'$, and, in particular, $p \notin G$, which again contradicts the choice of p. Thus $J \notin S$.

Since (I, S) locally models T, it follows that

$$(I, S \cup \{J\}, S) \not\models_{mbnf} T. \tag{9}$$

We claim, furthermore, that, for each PL-formula φ,

$$(I, S', S) \models_{mbnf} \varphi \text{ iff } (I, S \cup \{J\}, S) \models_{mbnf} \varphi. \tag{10}$$

This will be proved by induction on φ. First, let φ be a protected literal. If φ has the form *not* p or *not* $\neg p$ for an atom p, then (10) is obvious, because the possible worlds for *not* in both structures coincide. Let φ be $\mathrm{B}p$. If $(I, S', S) \models_{mbnf} \mathrm{B}p$, then $p \in H$ for each $H \in S'$, and, in particular, for each $H \in S$. Then, according to the definition of J, $p \in J$. Hence $(G, S \cup \{J\}, S) \models_{mbnf} \mathrm{B}p$. Conversely, if $(G, S \cup \{J\}, S) \models_{mbnf} \mathrm{B}p$, then $p \in H$ for each $H \in S \cup \{J\}$. Then, according to the definition of J, $p \in H$ for each $H \in S'$, so that $(I, S', S) \models_{mbnf} \mathrm{B}p$. Now let φ be $\mathrm{B}\neg p$. If $(I, S', S) \models_{mbnf} \mathrm{B}\neg p$, then $p \notin H$ for each $H \in S'$, and, in particular, for each $H \in S$. Then, according to the definition of J, $p \notin J$. Hence

$(G, S \cup \{J\}, S) \models_{mbnf} \mathrm{B}\neg p$. Conversely, if $(G, S \cup \{J\}, S) \models_{mbnf} \mathrm{B}\neg p$, then $p \notin H$ for each $H \in S \cup \{J\}$. Then, according to the definition of J, $p \notin H$ for each $H \in S'$, so that $(I, S', S) \models_{mbnf} \mathrm{B}\neg p$. The induction step is trivial if the main symbol of the formula is a propositional connective. In the case when the main symbol is B, it is sufficient to observe that $(I, S', S) \models_{mbnf} \mathrm{B}\varphi$ is equivalent to $(I, S', S) \models_{mbnf} \varphi$, and $(I, S \cup \{J\}, S) \models_{mbnf} \mathrm{B}\varphi$ is equivalent to $(I, S \cup \{J\}, S) \models_{mbnf} \varphi$, because φ is modalized and S is nonempty. When the main symbol is *not*, the reasoning is similar, using the fact that S' is nonempty (because it is a superset of S). This concludes the proof of (10).

It remains to observe now that, from (9) and (10),

$$(I, S', S) \not\models_{mbnf} T,$$

which contradicts the choice of S'. □

A few more lemmas are needed in order to establish the main theorem.

Lemma 4.1 *For any propositional combination φ of protected literals, any interpretation I and any nonempty set of interpretations S, $(I, S) \models_{rae} \varphi^r$ if and only if $(I, S \cup \{I\}, S) \models_{mbnf} \varphi$.*

Proof. Clearly, it is sufficient to prove the statement of the lemma for protected literals. Case 1: φ is $\mathrm{B}\psi$, where ψ is a literal. Then φ^r is $\mathrm{B}\psi$ also. Each of the conditions $(I, S) \models_{rae} \mathrm{B}\psi$, $(I, S \cup \{I\}, S) \models_{mbnf} \mathrm{B}\psi$ means that the literal ψ is true in all interpretations from $S \cup \{I\}$. Case 2: φ is *not* ψ, where ψ is a literal. Then φ^r is $\mathrm{B}\neg\mathrm{B}\psi$. By Proposition 2.3(b), $(I, S) \models_{rae} \mathrm{B}\neg\mathrm{B}\psi$ if and only if ψ is false in some interpretation from S, which is equivalent to $(I, S \cup \{I\}, S) \models_{mbnf}$ *not* ψ. □

Lemma 4.2 *For any propositional combination φ of protected literals, any interpretation I and any nonempty set of interpretations S, $(I, S) \models_{ae} \varphi^a$ if and only if $(I, S) \models_{rae} \varphi^r$.*

Proof. Clearly, it is sufficient to prove the statement of the lemma for protected literals. Case 1: φ is $\mathrm{B}\psi$, where ψ is a literal. Then φ^a is $\psi \wedge \mathrm{B}\psi$ and φ^r is $\mathrm{B}\psi$, so that the assertion of the lemma follows from Proposition 2.3(a). Case 2: φ is *not* ψ, where ψ is a literal. Then φ^a is $\neg\mathrm{B}\psi$ and φ^r is $\mathrm{B}\neg\mathrm{B}\psi$, so that the assertion of the lemma follows from Proposition 2.3(b). □

Lemma 4.3 *Let M, M' be sets of literals. If M is consistent and*

$$Th(Mod(M)) = Th(Mod(M')),$$

then $M = M'$.

Proof. Clearly, M' is consistent also. A literal φ belongs to $Th(Mod(M))$ if and only if it is a logical consequence of M, which is equivalent to $\varphi \in M$; similarly for M'. □

Recall that our goal is to prove the following fact:

Main Theorem. *Let T be a set of propositional combinations of protected literals. For any interpretation I and any nonempty set of interpretations S, the following conditions are equivalent:*

(i) (I, S) is an MBNF-model of T,

(ii) S is an autoepistemic model of T^a,

(iii) S is a reflexive autoepistemic model of T^r.

Moreover, for any consistent set M of literals, the following conditions are equivalent:

(iv) M is an answer set of T,

(v) $Th(Mod(M))$ is a stable expansion of T^a,

(vi) $Th(Mod(M))$ is a reflexive expansion of T^r.

Moreover, each consistent stable (reflexive) expansion of T^a (of T^r) has the form $Th(Mod(M))$ for some consistent set M of literals.

Proof. Let T be a set of propositional combinations of protected literals, I an interpretation, and S a nonempty set of interpretations. We will show first that conditions (i) and (iii) are equivalent. By the main lemma, (i) can be stated as the conjunction of two conditions:

(a) $(I, S, S) \models_{mbnf} T$,

(b) for each $J \notin S$, $(I, S \cup \{J\}, S) \not\models_{mbnf} T$.

On the other hand, (iii) is expressed by the equation

$$S = \{J : \text{for each } \varphi \in T^r,\ (J, S) \models_{rae} \varphi\},$$

which can be stated as the conjunction of two conditions:

(c) for each $J \in S$ and each $\varphi \in T$, $(J, S) \models_{rae} \varphi^r$,

(d) for each $J \notin S$ there is $\varphi \in T$ such that $(J, S) \not\models_{rae} \varphi^r$.

By Lemma 4.1, (c) is equivalent to the condition: For each $J \in S$,

$$(J, S, S) \models_{mbnf} T.$$

Since S is nonempty and all formulas in T are modalized, this is equivalent to (a). Furthermore, by Lemma 4.1, (d) is equivalent to the condition: For

each $J \not\in S$, there is $\varphi \in T$ such that $(J, S \cup \{J\}, S) \not\models_{mbnf} \varphi$. Since all formulas in T are modalized, this is equivalent to (b).

The fact that (ii) is equivalent to (iii) immediately follows from Lemma 4.2.

Let M be a consistent set of literals. By Proposition 2.1, condition (v) is equivalent to the condition: $Th(Mod(M)) = Th(S)$ for some nonempty autoepistemic model S of T^a. Using the equivalence of (i) and (ii) and Proposition 2.4, this can be further reformulated as follows: $Th(Mod(M)) = Th(Mod(M'))$ for some answer set M' of T. By Lemma 4.3, the equality $Th(Mod(M)) = Th(Mod(M'))$ is equivalent to $M = M'$, so that we can conclude that (v) is equivalent to (iv). For condition (vi) the proof is similar, with Proposition 2.2 used instead of Proposition 2.1.

Now let E be a consistent stable expansion of T^a. By Proposition 2.1, $E = Th(S)$ for some nonempty autoepistemic model of T^a. Using the equivalence of (i) and (ii), we conclude that (I, S) is an MBMF-model of T. By Proposition 2.4, it follows that $S = Mod(M)$ for some answer set M of T. For reflexive expansions of T^r, the proof is similar, with Proposition 2.2 and the equivalence of (i) and (iii) used, instead of Proposition 2.1 and the equivalence of (i) and (ii).

□

Acknowledgements

We are grateful to Michael Gelfond, Wiktor Marek, Norman McCain, Mirosław Truszczyński and Thomas Woo for useful discussions and for comments on a draft of this paper. Jianhua Chen and Mirosław Truszczyński have sent us drafts of the closely related papers [3] and [16]. This work was partially supported by National Science Foundation under grant IRI-9101078.

Notes

1. In [18], this operator is denoted by L.

2. Inserting B in front of every literal in the body of the rule, but not in the head [15], is another idea that may first seem promising. But if we apply it to the trivial rule $P \leftarrow P$, the result will be the axiom $\mathrm{B}P \supset P$, which has two stable expansions.

3. MBNF was developed as a generalization to the full predicate language of the system GK, proposed by Lin and Shoham [13]. The propositional fragment of MBNF is essentially equivalent to GK, as long as nested modalities are not involved.

4. Propositional combinations of protected literals are more general than disjunctive rules, because they may include positive occurrences of *not*. If φ is a propositional combination of protected literals in which *not* occurs only

negatively, then it can be written as a conjunction of (formulas corresponding to) disjunctive rules. If, as in [12], the language is assumed to include the logical constant "true", then the use of this constant in the scope of modal operators is another source of formulas that do not correspond to disjunctive rules.

5. This is reminiscent of the relationship between circumscription and pointwise circumscription [9].

References

[1] N. Bidoit and C. Froidevaux. Minimalism subsumes default logic and circumscription. In *Proceedings of LICS-87*, pages 89–97, 1987.

[2] N. Bidoit and C. Froidevaux. Negation by default and nonstratifiable logic programs. Technical Report 437, Université Paris XI, 1988.

[3] Jianhua Chen Minimal knowledge + negation as failure = only knowing (sometimes). In this volume.

[4] M. Gelfond. On stratified autoepistemic theories. In *Proceedings of AAAI-87*, pages 207–211, 1987.

[5] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In R. Kowalski and K. Bowen, editors, *Logic Programming: Proceedings of the Fifth International Conference and Symposium*, pages 1070–1080, 1988.

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

[7] M. Gelfond, V. Lifschitz, H. Przymusińska, and M. Truszczyński. Disjunctive defaults. In J. Allen, R. Fikes, and E. Sandewall, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, pages 230–237, 1991.

[8] H.J. Levesque. All I know: a study in autoepistemic logic. *Artificial Intelligence*, 42:263–309, 1990.

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

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

[11] V. Lifschitz. Minimal belief and negation as failure. Submitted for publication, 1992.

[12] V. Lifschitz and T.Y.C. Woo. Answer sets in general nonmonotonic reasoning. In *Principles of Knowledge Representation and Reasoning*, San Mateo, CA, 1992. Morgan Kaufmann. To appear.

[13] F. Lin and Y. Shoham. A logic of knowledge and justified assumptions. *Artificial Intelligence*, 57:271–290, 1992.

[14] W. Marek, G.F. Shvarts, and M. Truszczyński. Modal nonmonotonic logics: ranges, characterization, computation. Technical Report 187-91, Department of Computer Science, University of Kentucky, 1991. A revised version is to appear in the *Journal of ACM*.

[15] W. Marek and V.S. Subrahmanian. The relationship between logic program semantics and non-monotonic reasoning. In G. Levi and M. Martelli, editors, *Logic Programming: Proceedings of the Sixth International Conference*, pages 600–617, 1989.

[16] W. Marek and M. Truszczyński. The modal nonmonotonic logic of negation as failure. In this volume.

[17] R.C. Moore. Possible-world semantics autoepistemic logic. In R. Reiter, editor, *Proceedings of the workshop on non-monotonic reasoning*, pages 344–354, 1984. (Reprinted in: M. Ginsberg, editor, *Readings on nonmonotonic reasoning*. pages 137–142, 1990, Morgan Kaufmann.).

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

[19] G.F. Schwarz. Autoepistemic logic of knowledge. In W. Marek, A. Nerode and V.S. Submarahmanian, editors, *Logic programming and non-monotonic reasoning. Proceedings of the First International Workshop*, pages 260–274, Cambridge, MA, 1991. MIT Press.

[20] G.F. Schwarz. Minimal model semantics for nonmonotonic modal logics. In *Proceedings of LICS-92*, pages 34–43, 1992.