# Reflexive autoepistemic logic and logic programming

**V. Wiktor Marek**
**Mirosław Truszczyński**
Department of Computer Science
University of Kentucky
Lexington, KY 40506–0027
marek@ms.uky.edu, mirek@ms.uky.edu

## Abstract

In this paper we show that reflexive autoepistemic logic of Schwarz is a particularly convenient modal formalism for studying properties of answer sets for logic programs with classical negation and disjunctive logic programs. Syntactical properties of logic programs imply that a natural interpretation of default logic in the logic of minimal knowledge (nonmonotonic **S4F**) provides also a modal representation of logic programs. Moreover, in the case of logic programs one can use reflexive autoepistemic logic which is stronger and possesses simpler semantical characterizations than the logic of minimal knowledge. Reflexive autoepistemic logic and autoepistemic logic are bi-interpretable. Consequently, our results provide embeddings of logic programs with classical negation and disjunctive programs in autoepistemic logic.

## 1 Introduction

One of the problems driving recent investigations in logic programming is to provide a declarative account of logic programs. Two main problems arise. First, logic program clauses are rules that allow us to compute the head assuming that all conjuncts in the body have already been computed. Hence, they behave as inference rules rather than material implications. Secondly, the negation of $p$ in logic programming is treated as the inability of a program to prove $p$ rather than the falsity of $p$, and is often referred to as *negation as failure (to prove)*. Clearly, the inability of a program to prove $p$ does not mean that $\neg p$ is true. Hence, the classical interpretation of negation is inappropriate.

A significant amount of research on the semantics of the negation-as-failure operator originated from an observation that this form of negation behaves similarly to a modal operator *not provable*. As a result, the relationship between logic programming and modal logics has been studied extensively. The idea is to find a modal formalism whose interpretation of modal formulas would be well-suited for modeling the inference-rule nature of logic program clauses as well as negation-as-failure operator. The semantics of this formalism could then be adapted (translated) to the case of logic programs.

This general approach resulted in a spectacular achievement — the definition of a stable model of a logic program by Gelfond and Lifschitz [GL88]. The modal logic roots disappeared from their paper, but the definition was motivated by an embedding of logic programs into autoepistemic logic [Gel87, Gel90]. By means of this embedding the semantically defined concept of a stable expansion [Moo85] was adapted to the case of logic programs and yielded the class of stable models. In this fashion autoepistemic logic provided a declarative account of negation as failure.

As a result of this success we have witnessed a proliferation of modal logics proposed for modeling logic programs. These logics, usually patterned after autoepistemic logic, provided semantic justifications for several variants of negation, closely related but often different from the negation characterized by the stable model semantics ([Bon90, KM91, Prz91]).

Recently, two important extensions of logic programming have been proposed by Gelfond and Lifschitz. In [GL90] they proposed logic programs with classical negation in which clauses are built of literals rather than atoms (hence, classical negation is allowed). In addition, the negation-as-failure operator is applied to some of the literals in the body. To define the meaning of programs with classical negation Gelfond and Lifschitz introduced the notion of an *answer set*. Then, in [GL91], Gelfond and Lifschitz proposed an additional extension of the language by allowing *nonclassical* disjunctions in the heads. They called the resulting class of programs *disjunctive*. They extended the notion of an answer set from the case of programs with classical negation to the case of disjunctive programs. Gelfond and Lifschitz proved that answer sets coincide with stable models in the case of standard logic programs.

Both in the case of programs with classical negation and of disjunctive programs the notion of an answer set is defined in a procedural and not in a declarative fashion. An obvious attempt to find a declarative characterization, patterned after Gelfond's use of autoepistemic logic to characterize stable models, fails in this case. The original Gelfond interpretation of logic programs as autoepistemic theories [Gel87] can not be lifted to logic programs with classical negation (we discuss this issue in more detail below). The question whether logic programs with classical negation (and, more generally, disjunctive programs) can be embedded into a modal logic (in particular, autoepistemic logic) has been left open.

The notion of an answer set is based on two fundamental principles. First, clauses work as inference rules and serve the purpose of computing. Second, the interpretation of negation as failure is patterned after the principle of "jumping to conclusions subject to the lack of evidence to the contrary". Modal formalisms for answer sets must be capable of modeling both principles.

Autoepistemic logic is a logic of self-belief rather than knowledge. In particular, it allows cyclic arguments: believing in $\varphi$ justifies including $\varphi$ into a belief set. On the other hand, interpretation of clauses as inference rules does not allow cyclic arguments: the clause $p \leftarrow p$ does not justify the inclusion of $p$ into an answer set. Hence, the modality of autoepistemic logic cannot be used *directly* to capture computational character of clauses. It is easy to see that interpreting the rule $p \leftarrow p$ as the implication $Lp \supset p$ leads to an expansion containing $p$.

Default logic reflects both principles that we outlined above. Therefore, not surprisingly, logic programming with answer sets may be regarded as a fragment of (disjunctive) default logic [BF91, MT89, GL90, GLPT91].

Our approach to the problem of modal characterizations of logic programs builds on an earlier work [Tru91b, Tru91a, ST92]. In these papers the nonmonotonic logic **S4F** was proposed as *the* modal logic for (disjunctive) default reasoning. It follows, then, that logic programs can be interpreted within the nonmonotonic modal logic **S4F**. It is an interesting result because the nonmonotonic logic **S4F** is closely related with the minimal knowledge paradigm in knowledge representation ([HM85, Moo84, Lev90, LS90]). In particular, expansions in the nonmonotonic logic **S4F** have a preferred-model semantics ([Sch92, ST92]) which can be adapted easily to the case of answer sets. In this way a declarative description of answer sets can be provided.

Why then should we keep looking for better logics? There are at least two reasons. First, unlike in the case of autoepistemic logic, no propositional characterization of the nonmonotonic logic **S4F** is known so far. Secondly, the preference semantics of the nonmonotonic logic **S4F** is more complicated than that of autoepistemic logic.

But, can we find any better logic? It is known [ST92] that the nonmonotonic logic **S4F** is a maximal logic suitable for modeling default reasonings. However, the formalism of logic programs is syntactically simpler than that of default logic. Clauses of disjunctive logic programs are built of literals, disjunctions of literals are allowed in the heads, and not of arbitrary formulas as in the case of default logic. Because of this syntactic simplicity of logic programs we can do better than in the case of default logic.

In [Sch91], *reflexive autoepistemic logic* was proposed as an alternative to autoepistemic logic. This logic has all the attractive properties of the autoepistemic logic (several almost identical semantic characterizations of expansions) but defines the modality so that it models *knowledge* (which limits cyclic arguments) rather than *belief* (which allows them). Some applications of reflexive autoepistemic logic to logic programming have been mentioned in [Sch91] but its full potential has not been explored until now.

The main result of our paper shows an intuitively motivated and simple interpretation of clauses by modal formulas (in fact two interpretations) under which both logic programs with classical negation and disjunctive programs *can uniformly be embedded into reflexive autoepistemic logic.*

Reflexive autoepistemic logic is equivalent to autoepistemic logic. Specifically, there exist translations from each logic to the other one preserving the notion of expansion. Consequently, our embeddings of logic programs into reflexive autoepistemic logic yield the corresponding embeddings into autoepistemic logic. Autoepistemic logic interprets the modality as the operator of belief and not of knowledge. Speaking informally, the idea is to *simulate* the modality of knowing, needed to interpret logic programs, with the modality of belief available in autoepistemic logic. Once this is done, negation as failure can be described in autoepistemic logic as $\neg L$. A particularly appealing interpretation of logic programs in autoepistemic logic has been found by Lifschitz and Schwarz [LS93] and Chen [Che93] (see Section 4).

In the case of logic programs with classical negation (but without disjunctions in the heads) another, slightly different, embedding into autoepistemic logic is possible. Using this translation and a characterization result for autoepistemic expansions [MT91a] one gets a very elegant description of answer sets for such programs.

We hope that the reader will find in this paper arguments for our contention that reflexive autoepistemic logic is an appealing and powerful tool for studies of logic programming. Its capability to model properly both the rule character of logic programming clauses and also the negation as failure, coupled with an elegant semantics makes it a natural candidate for studies of semantical properties of logic programming with classical negation and disjunctive logic programs.

Our results as well as the results of [LS93, Che93] show that disjunctive logic programs can be embedded into autoepistemic logic. However, despite of this result and despite of the formal equivalence of reflexive autoepistemic and autoepistemic logics *it is reflexive autoepistemic logic and not autoepistemic logic that better reflects default logic roots of answer sets for logic programs*. It allows us to express logic programs as modal theories using (essentially) the same embedding that leads to a correct modal interpretation of default theories. Moreover, while the embedding into reflexive autoepistemic logic represents logic program clauses by clauses of the modal language, it is no longer true for the embedding into autoepistemic logic.

Due to size restrictions, this paper does not contain proofs of the results, In addition, we were not able to include two applications of the main result: a declarative description of answer sets, and the result showing that the formalism of nonmonotonic rule systems ([MNR90]) can be embedded (at a cost of introducing new atoms) into reflexive autoepistemic logic.

## 2 Modal interpretations of logic programming

In this section we will review past attempts at relating logic programming and modal nonmonotonic logics.

A *logic program* is a collection of *clauses* of the form

$$c \leftarrow a_1, \ldots, a_m, \mathbf{not}(b_1), \ldots, \mathbf{not}(b_n), \tag{1}$$

where all $a_i$, $b_i$ and $c$ are atoms. We will identify a program with the set of its all grounded Herbrand substitutions. Therefore, from now on, we restrict our attention only to propositional programs.

In our paper we also allow $a_i$, $b_i$ and $c$ to be literals. This yields a class of programs *with classical negation* [GL90]. We will also consider the case when $c = d_1 \sqcup \ldots \sqcup d_k$, where $d_i$'s are literals. The operator $\sqcup$ stands here for a nonstandard, "effective" disjunction. Programs with classical negation and with disjunctions of literals in the heads are called *disjunctive*. In [GL88, GL91] Gelfond and Lifschitz discussed the benefits of these extensions of logic programming for applications in knowledge representation, and introduced the concept of an *answer set* to specify the meaning of programs in these classes.

Let us recall the notion of an *answer set* for a disjunctive logic program $P$. A set of literals $S$ is *closed* under a disjunctive clause

$$d_1 \sqcup \ldots \sqcup d_k \leftarrow a_1, \ldots, a_m,$$

if for some $i$, $1 \leq i \leq k$, $d_i \in S$, or for some $i$, $1 \leq i \leq m$, $a_i \notin S$. Next, given a set of literals $S$ and an extended disjunctive logic program $P$, define the *reduct* of $P$ with respect to $S$ ($P^S$) to be the set of **not**-free clauses obtained from $P$ by removing each clause containing a literal $\mathbf{not}(a)$, where $a \in S$, and by removing all literals of the form $\mathbf{not}(a)$ from the remaining clauses. Finally, we say that $S$ is an answer set for $P$ if $S$ is a minimal set of literals such that

1. $S$ is closed under the rules in $P^S$,
2. $S$ is consistent or $S$ consists of all literals.

The definition of an answer set is procedural in its nature. Our goal in this paper is to find *declarative characterizations* of this notion. To this end we embed logic programs in a modal nonmonotonic logic.

As long as we are dealing with Horn programs (no negation as failure), there is little room for controversy. A clause

$$c \leftarrow a_1, \ldots, a_m$$

can be interpreted as:

$$a_1 \wedge \ldots \wedge a_m \supset c, \tag{2}$$

$$La_1 \wedge \ldots \wedge La_m \supset c, \quad \text{or} \tag{3}$$

$$La_1 \wedge \ldots \wedge La_m \supset Lc. \tag{4}$$

Some other interpretations are also possible. An important thing is that under all these interpretations no matter what modal logic contained in **S5** is used, the property of the existence of the least model of the Horn program in one way or another carries over to the modal case.

Now, a difficult part. What modality to use as an interpretation of **not**? And, what modal logic to select?

First attempts were made by Gelfond [Gel87] and Konolige [Kon88]. They interpret the fact that $p$ does not follow from a program as $\neg Lp$ ($p$ is not believed). When coupled with the interpretation (2) it yields the following modal formula for the clause (1):

$$a_1 \wedge \ldots \wedge a_m \wedge \neg Lb_1 \wedge \ldots \wedge \neg Lb_n \supset c \tag{5}$$

If the interpretation (3) is used, we get

$$La_1 \wedge \ldots \wedge La_m \wedge \neg Lb_1 \wedge \ldots \wedge \neg Lb_n \supset c \tag{6}$$

as a modal image of (1).

The nonmonotonic nature of the operator **not** requires us to use a modal nonmonotonic logic as means of reasoning from modal images of programs. Both interpretations (5) and (6) were studied in the context of autoepistemic logic. Gelfond [Gel87, Gel90] proved that the interpretability of Horn programs in autoepistemic logic can be lifted, via the translation (5), to the case of programs with **not** and yields the notion of a stable model. Specifically, Gelfond proved that $M$ is a stable model of a program $P$ if and only if $M$ is the set of atoms contained in a stable expansion of the image of $P$ under (5).

This approach does not work for any of the two extensions of logic programs mentioned earlier (classical negation, disjunctions in the heads).

**Example 2.1** Let $P = \{a \leftarrow b,\ \neg a \leftarrow\}$. Then, the theory $I = \{b \supset a,\ \neg a\}$ is the modal image of $P$ under the translation (5). $P$ has exactly one answer set: $\{\neg a\}$. The theory $I$ has one autoepistemic expansion, but it contains $\neg b$ in addition to $\neg a$, as well. So, if classical negation is allowed, Gelfond's approach fails even if **not** does not appear in a program.

Now, consider the disjunctive program $P = \{a \sqcup b \leftarrow\}$. It has two answer sets: $\{a\}$ and $\{b\}$. On the other hand, if a standard interpretation of disjunction is used, that is $a \vee b$, then the modal image of $P$, the theory $\{a \vee b\}$ has exactly one expansion generated by $a \vee b$ and containing neither $a$ nor $b$. $\Box$

The interpretation (6) has been considered in two contexts. In [MS89, MT91b] it is shown that embedding programs into autoepistemic logic using the translation (6) yields the concept of a supported model (and not the stable one). Moreover, this correspondence carries over to the class of programs with classical negation. Secondly, the interpretation (6) has been used in an early efforts to embed default logic in autoepistemic logic [Kon88].

A clause (1) can also be given a default interpretation as the default

$$\frac{a_1 \wedge \ldots \wedge a_m : M\neg b_1, \ldots, M\neg b_n}{c}. \tag{7}$$

This embedding is faithful both in the case of "standard" logic programs and programs with classical negation [BF91, MT89, GL90] and a similar embedding into the disjunctive default logic exists in the case of disjunctive programs [GLPT91].

Default logic can be embedded in the nonmonotonic logic **S4F** (see [MST91] for the definition of this and other modal logics considered in this paper) by means of each of the following two interpretations ([Tru91b, Tru91a]:

$$\frac{\varphi : M\beta_1, \ldots, M\beta_n}{\gamma} \quad \mapsto \quad L\varphi \wedge LM\beta_1 \wedge \ldots \wedge LM\beta_n \supset \gamma \tag{8}$$

$$\frac{\varphi : M\beta_1, \ldots, M\beta_n}{\gamma} \quad \mapsto \quad L\varphi \wedge LM\beta_1 \wedge \ldots \wedge LM\beta_n \supset L\gamma \tag{9}$$

As a corollary, we obtain that answer sets of logic programs with classical negation (hence, also stable models of "standard" logic programs) can be described as expansions in the nonmonotonic **S4F**. One has to use any of the following two interpretations of a clause (1):

$$La_1 \wedge \ldots \wedge La_m \wedge LM\neg b_1 \wedge \ldots \wedge LM\neg b_n \supset c \tag{10}$$

or

$$La_1 \wedge \ldots \wedge La_m \wedge LM\neg b_1 \wedge \ldots \wedge LM\neg b_n \supset Lc \tag{11}$$

In addition, a variant of the interpretation (11):

$$La_1 \wedge \ldots \wedge La_m \wedge LM\neg b_1 \wedge \ldots \wedge LM\neg b_n \supset Ld_1 \vee \ldots \vee Ld_k, \tag{12}$$

providing a modal image for a disjunctive clause with the head $d_1 \sqcup \ldots \sqcup d_k$, leads to a characterization of answer sets for disjunctive programs.

The main goal of this paper is to show that slightly modified versions of the embeddings (10) - (12) uniformly embed logic programming and logic programming with classical negation into reflexive autoepistemic logic, which has much simpler characterizations than the nonmonotonic logic **S4F**. Moreover, we will show that a versions of (12) provides a uniform modal interpretation in reflexive autoepistemic logic for all three classes of logic programs considered here: "standard" logic programs, logic programs with classical negation and disjunctive logic programs.

## 3 Reflexive autoepistemic logic and logic prog-rams

Reflexive autoepistemic logic was introduced by Schwarz [Sch91]. It assigns to a modal theory $I$ theories called *reflexive expansions*, which describe knowledge sets one can construct on the basis of $I$. Formally, a modal theory $T$ is a *reflexive expansion* of $I$ if

$$T = Cn(I \cup \{\varphi \equiv L\varphi : \varphi \in T\} \cup \{\neg L\varphi : \varphi \notin T\}). \tag{13}$$

One should note a close analogy with the definition of autoepistemic expansions [Moo85]: $T$ is an *autoepistemic expansion* of $I$ if

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

The main difference between these two logics is that, for $\varphi \in T$, autoepistemic logic uses $L\varphi$ as a premise in the process of reasoning, whereas reflexive autoepistemic logic uses the equivalence $\varphi \equiv L\varphi$. Hence, in reflexive autoepistemic logic if a formula $\varphi$ is assumed to be known then $\varphi$ and $L\varphi$ have the same logical value. This means that the modality is treated as "is known" rather than "is believed". It is also important to note a similarity of reflexive autoepistemic logic and the modal logic described by Przymusinski [Prz91]. The logic defined in [Prz91] also satisfies the requirement that $\varphi$ and $L\varphi$ be equivalent. The difference is that in [Prz91] $GCWA$ is used for generating negative information whereas here CWA with respect to modal atoms is used.

It turns out that reflexive autoepistemic logic is closely related to the modal logic **SW5**. We will recall now the definition of the logic **SW5**. The reader is referred to [HC84] for the detailed exposition of the concepts in modal logics that we use in our discussion.

A Kripke model $\mathcal{M} = \langle M, R, V\rangle$ (where, as usual, $M$ stands for a nonempty set of worlds, $R$ denotes an accessibility relation on worlds and $V$ assigns to each world a propositional valuation) is an **SW5**-*model* if $R = M \times M$ or $R = \{(a,a)\} \cup ((\{a\} \cup M) \times M)$, for some $a \notin M$.

The notions of satisfiability, $\langle \mathcal{M}, b\rangle \models \varphi$ and $\mathcal{M} \models \varphi$, are defined in a standard way. *The logic determined by the class of* **SW5**-*models is called the logic* **SW5**. Once the logic is defined, one can also define the corresponding notion of *entailment*, $I \models \varphi$.

It is not hard to see that the same logic is defined if we require that the valuations assigned to the worlds in $M$ are different. Each **SW5**-model can, hence, be represented by a singleton $\langle V\rangle$ or a pair $\langle v, V\rangle$, where $V$ is a set of propositional valuations representing valuations in the worlds of $M$, and $v$ is a propositional valuation in the world $a$. From now on we assume that **SW5**-models are of this form.

The semantic definition of **SW5** has a proof-theoretic counterpart. The logic **SW5** can equivalently be defined as the normal modal logic based on the axioms of the modal logic **S4** and the following consequence of the axiom 5:

W5: $\quad \neg L\neg L\varphi \supset (\varphi \supset L\varphi)$.

With each modal logic one can associate its nonmonotonic variant. The method was introduced in [MD80, McD82] and investigated in detail in [MST91]. The key notion here is that of an expansion. Given a modal logic $\mathcal{S}$, we define a modal theory $T$ to be an $\mathcal{S}$-*expansion* of a modal theory $I$ if

$$T = Cn_{\mathcal{S}}(I \cup \{\neg L\varphi : \varphi \notin T\}). \tag{14}$$

**Theorem 3.1 (Schwarz [Shv90, Sch91])** *Let $T$ be a propositionally consistent modal theory. For every theory $I$:*

1. *the theory $T$ is an autoepistemic expansion of $I$ if and only if $T$ is a* **KD45**-*expansion of $I$;*
2. *the theory $T$ is a reflexive expansion of $I$ if and only if $T$ is an* **SW5**-*expansion of $I$.* □

Since logic **KD45** has a similar semantic characterization to **SW5** (the only difference being that the world $a$ is not reflexive), this result points to more analogies between autoepistemic and reflexive autoepistemic logics. Moreover, the presence of the axiom T in **SW5** and its absence from **KD45** is an additional indication that autoepistemic logic interprets its modality as "is believed" while **SW5** interprets it as "is known".

The nonmonotonic logic **SW5**, and hence reflexive autoepistemic logic, can be characterized in terms of most preferred Kripke models. An **SW5**-model $\mathcal{V} = \langle V \rangle$ is *most preferred for a theory $I$* if

1. $\mathcal{V} \models I$; and
2. for every valuation $v$, if $\langle v, V \rangle \models I$ then $v \in V$.

Informally, $\mathcal{M}$ is most preferred if it cannot be extended by adding a new, essentially different, world in front of the cluster $V$.

**Theorem 3.2 (Schwarz [Sch92])** *A consistent theory $T$ is an* **SW5**-*expansion of a theory $I$ if and only if $T$ is the theory of a most preferred model for $I$.* □

This characterization of **SW5** establishes one more similarity with the autoepistemic logic which was described in analogous terms in [Moo84].

Our first theorem illustrates this point by showing that the concept of the answer set can be modeled within reflexive autoepistemic logic. To this end, we introduce the following two intuitive modal encodings of a clause (1):

$$La_1 \wedge \ldots \wedge La_m \wedge L\neg Lb_1 \wedge \ldots \wedge L\neg Lb_n \supset c \tag{15}$$

and, in the case of a disjunctive clause when $c = d_1 \sqcup \ldots \sqcup d_k$,

$$La_1 \wedge \ldots \wedge La_m \wedge L\neg Lb_1 \wedge \ldots \wedge L\neg Lb_n \supset Ld_1 \vee \ldots \vee Ld_k. \tag{16}$$

This latter interpretation can also be applied for programs without disjunctions (when $c = d_1$).

Both translations are quite intuitive. For example, (16) can be read as:

> If for every $i$, $1 \leq i \leq m$, $a_i$ is known and, for every $i$, $1 \leq i \leq n$, it is known that $b_i$ is not known ($\neg b_i$ is possible), then at least one $d_i$ is known.

That is, we interpret a clause as an inference rule which, in order to be applied has to have all its premises established (all premises have to be known). To achieve this effect the modal atom $La_i$ appears in the antecedent of the modal translation. Similarly, we need modal atoms to express that we know about $\mathbf{not}(b_i)$. Since $\mathbf{not}(b_i)$ can be read as "$b_i$ not known", the fact that the premise $\mathbf{not}(b_i)$ is known is expressed as $L\neg Lb_i$.

In what follows we will focus on formulas of the form:

$$La_1 \wedge \ldots \wedge La_m \wedge L\neg Lb_1 \wedge \ldots \wedge L\neg Lb_n \supset Ld_1 \vee \ldots \vee Ld_k, \qquad (17)$$

where all $a_i, b_i$ and $d_i$ are literals. We will call such formulas *lp*-clauses. Hence, *lp*-clauses are variants of the interpretation (12). The difference is that when we eliminate the operator $M$ by means $\neg L \neg$ we also reduce the double negation of $b_i$.

Let us observe that an *lp*-clause, that is, a formula of the form (17), is valid in (i.e. true in every world of) a model $\langle v, V\rangle$ if and only if the following two conditions hold:

**(LP1)** whenever all $a_i$, $1 \leq i \leq m$, are true in all valuations of $\{v\} \cup V$ and, for each $i$, $1 \leq i \leq n$, $b_i$ is not true in at least one valuation from $V$, then at least one $d_i$ is true in all valuations of $\{v\} \cup V$;

**(LP2)** whenever all $a_i$, $1 \leq i \leq m$, are true in all valuations from $V$ and, for each $i$, $1 \leq i \leq n$, $b_i$ is not true in at least one valuation from $V$, then at least one $d_i$ is true in all valuations of $V$.

Using this observation we will prove now two simple properties of most preferred models of theories consisting of *lp*-clauses.

**Proposition 3.3** *Let $I$ consist of lp-clauses. Let $\langle V\rangle$ be a most preferred model for $I$ and let $M$ be the set of all literals true in all valuations in $V$. Then, for every valuation $w$, if all the literals in $M$ are true in $w$, then $w \in V$.*

**Proposition 3.4** *Let $I$ consist of lp-clauses. Let $\langle V\rangle$ be a most preferred model of $I$. Then for every model $\langle V'\rangle$ of $I$, if $V \subseteq V'$ then $V = V'$.*

A theory $T$ in the modal language is *stable* if it is closed under propositional provability and is closed under positive and negative introspection.

That is, for every $\varphi \in T$, $L\varphi \in T$ (positive introspection), and for every $\varphi \notin T$, $\neg L\varphi \in T$ (negative introspection). It is well known that a modal-free theory $S$ there exists a unique stable theory $T$ such that the modal-free part of $T$ is $Cn(S)$ ([Moo85]). We will denote this theory by $ST(S)$.

We have the following theorem establishing adequacy of reflexive autoepistemic logic for logic programming applications.

**Theorem 3.5** *Let $S \subseteq \mathcal{L}$ be a consistent set of literals. Then, $S$ is an answer set for a disjunctive logic program $P$ if and only if $ST(S)$ is a reflexive expansion for the image of $P$ under (16).*

For a translation (15) we obtain a similar result.

**Theorem 3.6** *Let $S$ be a consistent set of literals. Let $P$ be a logic program with classical negation (no disjunctions in the heads). Then $S$ is an answer set for $P$ if and only if $ST(S)$ is a reflexive autoepistemic expansion for the image of $P$ under translation (15).* □

# 4 Answer sets and autoepistemic logic

In Section 2 we noticed that although the stable semantics for logic programs can be faithfully represented in autoepistemic logic (nonmonotonic **KD45**), a similar result for logic programs with classical negation and for disjunctive programs cannot be obtained by a simple extension of Gelfond's translation. In this section we develop a technique to embed disjunctive logic programs (and, in particular, logic programs with classical negation) into autoepistemic logic. These embeddings connect answer sets and autoepistemic expansions. Our line of reasoning uses a mutual interpretability result for nonmonotonic **KD45** and **SW5** [Sch91]. We will be interested here in the interpretation of **SW5** in **KD45**.

**Definition 4.1** For every modal formula $\varphi$ we recursively define the formula $\varphi_B$:

1. $(p)_B = p$, for every atom $p$;
2. $(\neg\varphi)_B = \neg(\varphi_B)$;
3. $(\varphi \circ \psi)_B = \varphi_B \circ \psi_B$, where $\circ$ stands for a binary boolean connective;
4. $(L\varphi)_B = \varphi_B \wedge L(\varphi_B)$.

For a theory $I$ we define $I_B = \{\varphi_B : \varphi \in I\}$. □

The following result connects nonmonotonic **SW5** and nonmonotonic **KD45** (that is autoepistemic logic).

**Proposition 4.1 (Schwarz [Sch91])** *Let $I \subseteq \mathcal{L}_L$. Then for every consistent stable theory $T$, $T$ is an* **SW5***-expansion of $I$ if and only if $T$ is an* **KD45***-expansion of $I_B$.* $\Box$

The transformation $(\cdot)_B$ treats "knowledge" as "true belief", and has been investigated before, for instance in the context of provability logics, see [Smo85] for further details. For the sake of completeness let us mention that there exists a translation converse to $(\cdot)_B$, with similar properties.

Let us compose our embedding (16) of (disjunctive) logic programs with the translation $(\cdot)_B$ and use properties of logic **KD45**. As a result, we get a translation of logic programming clauses into the modal language (found by Lifschitz and Schwarz [LS93], and Chen [Che93]):

$$(a_1 \wedge La_1) \wedge \ldots \wedge (a_m \wedge La_m) \wedge \neg Lb_1 \wedge \ldots \wedge \neg Lb_n \supset (d_1 \wedge Ld_1) \vee \ldots \vee (d_k \wedge Ld_k). \tag{18}$$

In particular, we obtain an alternative argument for the following theorem of Lifschitz and Schwarz [LS93].

**Theorem 4.2** *Let $P$ be a disjunctive logic program and $S$ a consistent set of literals. Then $S$ is an answer set for $P$ if and only if $ST(S)$ is an autoepistemic expansion of the theory obtained from $P$ by applying the translation (18).* $\Box$

We will compare now the embedding (16) into reflexive autoepistemic logic and the embedding (18) into autoepistemic logic. The first of them transforms a disjunctive program clause into a *clause*, that is, (up to a simple propositional logic transformation) a disjunction of modal literals. The embedding (18) does not share this property. A formula (18) is not a clause. It can be represented as a conjunction of clauses consisting of modal and propositional literals. Such a transformation is, however, very expensive. The reason is that the formula in the head:

$$(d_1 \wedge Ld_1) \vee \ldots \vee (d_k \wedge Ld_k)$$

generates $2^k$ clauses:

$$d_{j_1} \vee \ldots \vee d_{j_r} \vee Ld_{i_1} \vee \ldots \vee Ld_{i_s},$$

where $\{i_1, \ldots, i_s\} \cap \{j_1, \ldots, j_r\} = \emptyset$ and $\{i_1, \ldots, i_s\} \cup \{j_1, \ldots, j_r\} = \{1, \ldots, k\}$. Therefore every formula of the form (18) requires $2^k$ clauses of the form:

$$a_1 \wedge La_1 \wedge \ldots a_n \wedge La_m \wedge \neg Lb_1 \wedge \ldots \wedge \neg Lb_n \supset d_{j_1} \vee \ldots \vee d_{j_r} \vee Ld_{i_1} \vee \ldots \vee Ld_{i_s}.$$

Hence, the size of the clausal representation of translation (18) may be exponential.

In the case of programs without disjunction in the heads just two clauses suffice. However, a particularly elegant embedding of logic programs with classical negation into autoepistemic logic is obtained when we use the translation (15) instead of (16). That is when we do *not* have $Lc$ in the head but put $c$ instead. In this fashion the clause

$$c \leftarrow a_1, \ldots, a_m, \mathbf{not}(b_1), \ldots \mathbf{not}(b_n)$$

is expressed by

$$La_1 \wedge \ldots \wedge La_m \wedge L\neg Lb_1 \wedge \ldots \wedge L\neg Lb_n \supset c.$$

When we combine this translation with the translation $(\cdot)_B$ of reflexive autoepistemic logic into autoepistemic logic and then apply axioms of **KD45** for simplifying modalities, we get:

$$a_1 \wedge La_1 \wedge \ldots \wedge a_n \wedge La_m \wedge \neg Lb_1 \wedge \ldots \wedge \neg Lb_n \supset c$$

This formula, in turn, can be transformed in propositional logic into:

$$La_1 \wedge \ldots \wedge La_m \wedge \neg Lb_1 \wedge \ldots \wedge \neg Lb_n \supset (a_1 \wedge \ldots \wedge a_m \supset c) \qquad (19)$$

Thus we obtain the following result.

**Theorem 4.3** *Let $S$ be a consistent set of literals. Let $P$ be a logic program with classical negation. Let $I$ be the image of $P$ under translation (19). Then $S$ is an answer set for $P$ if and only if $ST(S)$ is an autoepistemic expansion of $I$.* □

The translation (19) provides a very clear illustration how autoepistemic logic can be used to represent inference-rule nature of clauses with classical negation (but without disjunctions). To represent a program clause

$$c \leftarrow a_1, \ldots, a_m, \mathbf{not}(b_1), \ldots \mathbf{not}(b_n)$$

in autoepistemic logic we have to require that positive premises are believed and negative ones are not. But these beliefs must imply a *weaker conclusion*. Namely, instead of $c$, we have the formula $a_1 \wedge \ldots \wedge a_m \supset c$ in the consequent of the implication. This formula can be used in the process of deriving $c$ but does not guarantee that $c$ will be actually derived.

As another application of Theorem 4.3 we obtain a clean characterization of answer sets for logic programs with classical negation. It is based on the notion of the strong reduct and refers to the "logic" interpretation of a **not**-free program clause

$$c \leftarrow a_1, \ldots, a_m,$$

as the implication

$$a_1 \wedge \ldots \wedge a_m \supset c.$$

We will define now the notion of the *strong $S$-reduct* of a program $P$. Recall that in the original reduct of a program [GL88, GL90], the clause

$$c \leftarrow a_1, \ldots, a_m, \mathbf{not}(b_1), \ldots \mathbf{not}(b_n)$$

is eliminated if and only if some $b_i$ does belong to $S$. In the strong $S$-reduct we eliminate more clauses. Specifically, we eliminate a clause also if some $a_i$ *does not* belong to $S$. From the remaining clauses we remove their negative part. That is, the strong $S$-reduct consists of those clauses $c \leftarrow a_1, \ldots, a_m$ of the reduct, for which $a_1, \ldots, a_m \in S$. We shall denote by $P_S$ the strong $S$-reduct of $P$.

Let us apply the characterization result for autoepistemic expansions [MT91a] to the image of a logic program $P$ under the translation (19). A formula of the form (19) has, as the objective part the formula

$$a_1 \wedge \ldots \wedge a_m \supset c.$$

But such formula will be used in the process of generating $ST(S)$ *only* if all $a_j$ belong to $S$, and *no* $b_i$ does. That is precisely when

$$c \leftarrow a_1, \ldots, a_m$$

belongs to the strong $S$-reduct of $P$! Consequently, the following result can be derived.

**Proposition 4.4** *Let $P$ be a logic program with classical negation and let $S$ be a consistent set of literals. Then, $S$ is an answer set for $P$ if and only if $S$ is the set of literals entailed by "logical" images of clauses from $P_S$, that is, by the set of formulas*

$$\{a_1 \wedge \ldots \wedge a_m \supset c \colon c \leftarrow a_1, \ldots, a_m \in P_S\}. \qquad \Box$$

It is important to note that the assertion of Proposition 4.4 fails for the original notion of reduct. The program of Example 2.1 is an illustration of this phenomenon.

# 5 Conclusions

We have shown that reflexive autoepistemic logic (nonmonotonic **SW5**) is a convenient and natural tool to study answer sets for disjunctive logic programs and logic programs with classical negation.

Our results provide embeddings of disjunctive logic programs into autoepistemic logic. One of these embeddings gives a very elegant characterization of answer sets of logic programs with classical negation (but with no disjunctions in heads).

Despite the fact that logic programs can be embedded into autoepistemic logic, we believe that it is reflexive autoepistemic logic that is particularly well suited for modal representations of logic programs. The embedding into reflexive autoepistemic logic is (essentially) the same as embedding of defaults into the modal language. Moreover, clauses of disjunctive logic programs are represented in this embedding as clauses of the modal language and not by "non-clausal" formulas as in the case of the embedding into autoepistemic logic.

## Acknowledgements

This work was partially supported by National Science Foundation under grant IRI-9012902. The authors gratefully acknowledge comments by Vladimir Lifschitz and Grigori Schwarz.

## References

[BF91] N. Bidoit and C. Froidevaux. Negation by default and unstratifiable logic programs. *Theoretical Computer Science*, 78:85–112, 1991.

[Bon90] P. Bonatti. A more general solution to the multiple expansion problem. In *Proceedings of the workshop on nonmonotonic reasoning and logic programming, North American Conference on Logic Programming*, 1990. Austin, Texas, to appear in *Methods of Logic in Computer Science*.

[Che93] J. Chen. Minimal knowledge + negation as failure = only knowing (sometimes). In A. Nerode and L. Pereira, editors, *Logic programming and non-monotonic reasoning*. MIT Press, 1993.

[Gel87] M. Gelfond. On stratified autoepistemic theories. In *Proceedings of AAAI-87*, pages 207–211, Los Altos, CA., 1987. American Association for Artificial Intelligence, Morgan Kaufmann.

[Gel90] M. Gelfond, 1990. Personal communication.

[GL88] M. Gelfond and V. Lifschitz. The stable semantics for logic programs. In R. Kowalski and K. Bowen, editors, *Proceedings of the 5th international symposium on logic programming*, pages 1070–1080, Cambridge, MA., 1988. MIT Press.

[GL90] M. Gelfond and V. Lifschitz. Logic programs with classical negation. In D. Warren and P. Szeredi, editors, *Proceedings of the 7th international conference on logic programming*, pages 579–597, Cambridge, MA., 1990. MIT Press.

[GL91] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.

[GLPT91] M. Gelfond, V. Lifschitz, H. Przymusinska, and M. Truszczyński. Disjunctive defaults. In *Second international conference on principles of knowledge representation and reasoning, KR '91*, Cambridge, MA, 1991.

[HC84] G.E. Hughes and M.J. Cresswell. *A companion to modal logic*. Methuen and Co. Ltd., London, 1984.

[HM85] J.Y. Halpern and Y. Moses. Towards a theory of knowledge and ignorance: preliminary report. In K. Apt, editor, *Logics and models of concurrent systems*, pages 459–476. Berlin: Springer-Verlag, 1985. NATO ASI Series F, Vol 13.

[KM91] A.C. Kakas and P. Mancarella. Negation as stable hypotheses. In A. Nerode, W. Marek, and V.S. Subrahmanian, editors, *Logic programming and non-monotonic reasoning*, pages 275–288. MIT Press, 1991.

[Kon88] K. Konolige. On the relation between default and autoepistemic logic. *Artificial Intelligence*, 35:343–382, 1988.

[Lev90] H. J. Levesque. All I know: a study in autoepistemic logic. *Artificial Intelligence*, 42:263–309, 1990.

[LS90] F. Lin and Y. Shoham. Epistemic semantics for fixed-points non-monotonic logics. In *Proceedings of TARK-90*, pages 111–120, San Mateo, CA., 1990. Morgan Kaufmann.

[LS93] V. Lifschitz and G. Schwarz. Extended logic programs as autoepistemic theories. In A. Nerode and L. Pereira, editors, *Logic programming and non-monotonic reasoning*. MIT Press, 1993.

[McD82] D. McDermott. Nonmonotonic logic II: nonmonotonic modal theories. *Journal of the ACM*, 29:33–57, 1982.

[MD80] D. McDermott and J. Doyle. Nonmonotonic logic I. *Artificial Intelligence*, 13:41–72, 1980.

[MNR90] W. Marek, A. Nerode, and J.B. Remmel. Nonmonotonic rule systems I. *Annals of Mathematics and Artificial Intelligence*, 1:241–273, 1990.

[Moo84] R.C. Moore. Possible-world semantics for 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.).

[Moo85] R.C. Moore. Semantical considerations on non-monotonic logic. *Artificial Intelligence*, 25:75–94, 1985.

[MS89] W. Marek and V.S. Subrahmanian. The relationship between logic program semantics and non-monotonic reasoning. In *Proceedings of the 6th international conference on logic programming*, pages 600–617, 1989.

[MST91] W. Marek, G.F. Shvarts, and M. Truszczyński. Modal nonmonotonic logics: ranges, characterization, computation. In *Proceedings of the 2nd international conference on principles of knowledge representation and reasoning, KR '91*, pages 395–404, San Mateo, CA., 1991. Morgan Kaufmann. An extended version of this article will appear in the Journal of the ACM.

[MT89] W. Marek and M. Truszczyński. Stable semantics for logic programs and default theories. In E.Lusk and R. Overbeek, editors, *Proceedings of the North American conference on logic programming*, pages 243–256, Cambridge, MA., 1989. MIT Press.

[MT91a] W. Marek and M. Truszczyński. Autoepistemic logic. *Journal of the ACM*, 38:588–619, 1991.

[MT91b] W. Marek and M. Truszczyński. Computing intersection of autoepistemic expansions. In A. Nerode, W. Marek, and V.S. Subrahmanian, editors, *Logic programming and non-monotonic reasoning*, pages 37–50. MIT Press, 1991.

[Prz91] T. Przymusinski. Autoepistemic logic of closed beliefs and logic programming. In A. Nerode, W. Marek, and V.S. Subrahmanian, editors, *Logic programming and non-monotonic reasoning*, pages 3–20. MIT Press, 1991.

[Sch91] G.F. Schwarz. Autoepistemic logic of knowledge. In A. Nerode, W. Marek, and V.S. Subrahmanian, editors, *Logic programming and non-monotonic reasoning*, pages 260–274. MIT Press, 1991.

[Sch92] G.F. Schwarz. Minimal model semantics for nonmonotonic modal logics. In *Proceedings of LICS-92*, 1992.

[Shv90] G.F. Shvarts. Autoepistemic modal logics. In R. Parikh, editor, *Proceedings of TARK 1990*, pages 97–109, San Mateo, CA., 1990. Morgan Kaufmann.

[Smo85] C. Smorynski. *Self-reference and modal logic*. Berlin: Springer-Verlag, 1985.

[ST92] G.F. Schwarz and M. Truszczyński. Modal logic **S4F** and the minimal knowledge paradigm. In *Proceedings of TARK 1992*, San Mateo, CA., 1992. Morgan Kaufmann.

[Tru91a] M. Truszczyński. Embedding default logic into modal nonmonotonic logics. In A. Nerode, W. Marek, and V.S. Subrahmanian, editors, *Logic programming and non-monotonic reasoning*, pages 151–165. MIT Press, 1991.

[Tru91b] M. Truszczyński. Modal interpretations of default logic. In *Proceedings of IJCAI-91*, pages 393–398, San Mateo, CA., 1991. Morgan Kaufmann.