Answer Sets and Nonmonotonic S4

David Pearce
Gruppe LWI
Institut für Philosophie, Freie Universität Berlin
Habelschwerdter Allee 30, D-14195 Berlin
pearce@inf.fu-berlin.de

Abstract. The semantics of *answer sets* ([10]) provides a flexible tool for interpreting various forms of extended logic programs and deductive databases. In particular, it is applicable to extended disjunctive databases whose rules may contain two kinds of negation as well as disjunctive conclusions. Extending our earlier work, [19, 20], we provide here a complete logical characterisation of answer set inference as a subsystem of nonmonotonic S4. The method consists in interpreting the rules of disjunctive databases as formulas of Nelson's constructive logic with strong negation, N. In particular, the two types of negation present in database rules are interpreted as Nelson's *strong* negation and Heyting's *intuitionistic* negation, respectively. We then make use of the well-known Gödel embedding of N to modal S4, and show that the inference relation associated with the answer set semantics is equivalent to that of nonmonotonic S4. As corollaries we obtain in N a monotonic lower-bound for answer set inference as well as the related modal embeddings recently established in [12] and [13].

1 Introduction

One of the most powerful and versatile approaches to extended logic programming has been the one developed by Michael Gelfond and Vladimir Lifschitz in [8, 9, 10, 7]. Starting from the format of negation-free program clauses or rules, they added successively: *weak* negation (in the style of negation-as-failure), [8]; *strong* negation (which they called 'classical'), [9]; *disjunctive* heads, [10]; and recently *epistemic* operators, [7]. Each system is equipped with a fixpoint semantics that conservatively extends its predecessor; the basic semantical units being called variously, stable models, answer sets or belief sets.

We shall focus here on disjunctive databases whose clauses are represented in [10] as *rules* of the form

$$K_1 | \ldots | K_k \leftarrow L_1, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n \qquad (1)$$

where the L_i, K_j are atoms or strongly negated atoms, the rule's 'body' is a

conjunction, its 'head' a disjunction, and '*not*' is a weak negation operator in the style of negation-as-failure.

In general, database rules like (1) are not interpreted as logical formulas and the answer sets of a database are characterised by a fixpoint definition, without reference to any underlying logical system. Even the notation is carefully chosen so as to minimise confusion with the customary logical connectives. Conceptually, this is rather far removed from the declarative ideal of 'programming in logic'. A further consequence is the fact that there is no accompanying proof theory.

One way to provide a logical interpretation and to restore, at least partially, the declarative ideal is to rewrite (1) as a formula of autoepistemic logic. This idea was first carried out by Gelfond for general logic programs whose rules (1) contain neither disjunction or nor strong negation. His embedding [6] into autoepistemic logic made the plausible assumption that weak negation, as 'failure-to-prove', can be regarded as a modal concept, so that the subformula '$notA$' of a program rule (where A is an atom) is translated by '$\sim BA$', where 'B' is a modal or doxastic belief operator. However, it proved difficult to extend Gelfond's transformation to cover the case where program rules contain arbitrary literals and disjunctive heads.

The question of how to relate the answer set semantics to a nonmonotonic modal system in the general case was recently independently addressed by Lifschitz & Schwarz [12], by Marek & Truszczynski [13] and, less directly, by Chen [4]. However, while Gelfond's original interpretation of weak negation was simple and plausible, the more recent embeddings that have been proposed are less transparent and might even appear to some as *ad hoc*. For instance, it is not entirely clear to what extent they can be regarded as extensions of the Gelfond translation and whether they provide natural interpretations of database rules. In particular, why do these embeddings interpret *all* the literals in a rule (and not only those prefixed by '*not*') modally?

In this paper the clauses of extended programs and disjunctive databases will be intepreted not as rules but as ordinary logical formulas; only the logic involved will be not classical but the constructive system N of Nelson [17]. The program arrow '$\leftarrow$', the 'explicit' negation '$\neg$' and the 'disjunction' '$|$' are simply read as the constructive implication '$\rightarrow$', the strong negation '$\sim$' and the disjunction '$\vee$', respectively, of N. In addition, I shall identify weak negation '*not*' with intuitionistic negation, '$-$', and use here the 'full' logic N in which '$-$' appears as a defined connective, [11]. This interpretation appears to be more adequate than the 'inference-rule' reading of program clauses, since the latter breaks down entirely when disjunction is present.

A further gain is that once database rules are rewritten as formulas of N we can take advantage of a standard embedding into modal logic, replacing the constructive connectives by classical ones, appropriately interspersed with modal operators. This embedding is just a simple variant of the Gödel translation of intuitionistic logic into S4. A constructive formula like $A \rightarrow B$, where A, B are atoms, becomes $\Box(\Box A \supset \Box B)$. This explains why even the atoms of a database

rule have to be prefixed by a modal operator. Since the strong negation of N is different from intuitionistic negation, the embedding of N into S4 is a proper extension of the Gödel embedding, giving a different interpretation to $\sim$. For example a strongly negated atom, $\sim A$, is mapped to $\Box \sim A$, whereas an intuitionistically negated atom, $-A$, is mapped to $\Box \sim \Box A$.

The Gödel translation τ is a natural and well-understood way of endowing the constructive connectives with a modal or epistemic reading; so the translation is certainly a natural one. Moreover, since τ is a recursive translation on all formulas, it works uniformly for logic programs and disjunctive databases alike; there is no need to deal with them as separate cases. The 'reflexive autoepistemic' translation of [13] is a special case of τ that is equivalent for the relevant formulas to be considered. So this translation is certainly not *ad hoc*, but just a simplified form of the Gödel one. The other, 'autoepistemic' translation of [12] seems neither especially natural nor interesting from the modal point of view, unless one lays special emphasis on working with nonreflexive frames. It achieves no more and no less than that.

2 Logical Preliminaries

2.1 Constructive Logic

We assume throughout a fixed, countable predicate language containing at least one individual constant or name. We denote by N constructive logic with strong negation, introduced by Nelson [17]. Terms and formulas are built-up in the usual manner, using the logical constants of N: $\wedge, \vee, \sim, -, \rightarrow, \exists, \forall$. The negation '$\sim$' is called *strong negation*, and in addition N contains a further *intuitionistic* negation '$-$', which can be defined in N by

$$-\varphi \equiv \varphi \rightarrow \sim\varphi.$$

Using this definition, N turns out to be a conservative extension of Heyting's intuitionistic logic H. Notice that Nelson's negation '$\sim$' is termed 'strong', since in the combined full system N, $\sim\varphi \rightarrow -\varphi$ is a theorem.

Literals are either atoms or strongly negated atoms; a literal with no free variables is called *ground*. The set of all ground literals is denoted by by *Lit*. For reasons of space we do not present here a formal system corresponding to N. The reader is referred instead to [11, 5], where axiom systems are presented. These are obtained from the standard axioms for intuitionistic predicate logic by adding new axioms governing strong negation, '$\sim$'. The derivability relation for N is denoted by $\vdash_N$. Gentzen-style sequent systems for N can be found in [3, 11] (cf. also [27, 28]). A tableau proof system for propositional N is discussed in [22]. A Kripke-style semantics (and completeness proof) for N is given in [11, 1], and for a slight variant of N in [26].

2.2 Answer Sets for Disjunctive Databases

We shall represent an *extended disjunctive database*, or XDDB for short, as a collection Π of formulas of the form

$$L_1 \wedge \ldots \wedge L_m \wedge -L_{m+1} \wedge \ldots \wedge -L_n \rightarrow K_1, \vee \ldots \vee K_k \qquad (2)$$

where each L_i, K_j is a literal, $k \geq 1$ and we may have $m = n$ and m or n may be zero. Notice that '$-$' only appears directly before a literal and to the left of the implication; all other negations in (2) are strong. An XDDB is a set of such formulas, and an extended logic program or XLP, Π, is a set of such formulas, with $k = 1$. Since answer sets are defined for databases without variables, each formula of form (2) is treated as shorthand for the set of its ground instances. This is only a notational variant of the formalism of Gelfond & Lifschitz [10], who regard database formulas as a rules of the form (1), and employ the symbols '.', '|', '$\leftarrow$' and '*not*' for conjunction, disjunction, implication and weak negation, respectively.

Our use of the ordinary logical connectives of N, instead of the special notation employed by Gelfond, Lifschitz and others, is justified by a completeness result (Proposition 1 below) for $-$-free databases, and is therefore quite uncontroversial in the monotonic case. For the full, nonmonotonic case, where also weak negation appears, the matter is more subtle. Since, under the answer set semantics described below, the presence of *not* renders database inference nonmonotonic, we cannot expect, in the general case, to obtain a correspondence between '*not*' and intuitionistic negation '$-$' within the ordinary constructive system N. Notice, however, that '*not*' is conceptually rather close to Heyting's intuitionistic negation in that both are forms of negation as non-provability. Accordingly, we shall interpret '*not*' as if it were simply intuitionistic negation, '$-$'. Subsequently, nonmonotonicity and thereby a genuine correspondence between the two negations will be restored when we come to define modal embeddings, since we shall then work not in ordinary but in nonmonotonic modal logic.

Let Π be a database without '$-$'. Adapting the definition of [10] to the above notation, we recall that an *answer set* of Π is a minimal (under set-theoretic inclusion) subset S of *Lit* such that

(A1) for each formula $L_1 \wedge \ldots \wedge L_m \rightarrow K_1 \vee \ldots \vee K_k$ of Π, if $L_1, \ldots, L_m \in S$ then, for some $i = 1, \ldots, k$, $K_i \in S$;

(A2) if S contains a pair of complementary literals of the form $A, \sim A$ (where A is an atom), then $S = Lit$.

Let Π be an XDDB. For any set of ground literals $S \subset Lit$, the database Π^S is the database without '$-$' obtaining from Π by deleting

(i) each formula containing a subformula $-L$ with $L \in S$, and

(ii) all subformulas of the form $-L$ in the remaining formulas.

Since the transformed database does not contain '$-$', its answer sets are defined (as previously). Then a set S of ground literals is said to be an *answer set* of an extended disjunctive database Π if and only if S is an answer set of Π^S.

We call an answer set S of Π *consistent* if it does not contain a complementary pair of literals. A database Π is said to be consistent if it does not possess an inconsistent answer set. Clearly, we can regard an atomic formula A as being true in an answer set S if $A \in S$ and false if $\sim A \in S$. In order to extend this semantics to Boolean compound statements we make use of a recent generalisation of answer sets proposed by Michael Gelfond [7]. Adapting Gelfond's approach to the case of XDDBs, truth ($\models$) and falsity ($\models\!\!\!|$) of formulas in $\mathcal{L} = \mathcal{L}(\wedge, \vee, \sim)$ wrt a set S of ground literals, is defined by:

$$\text{For atomic } A,\ S \models A \text{ if } A \in S;\ S \models\!\!\!| A \text{ if } \sim A \in S$$

$$S \models \sim A \text{ if } S \models\!\!\!| A;\ S \models\!\!\!| \sim A \text{ if } S \models A$$

$$S \models \varphi \wedge \psi \text{ if } S \models \varphi \text{ and } S \models \psi;\ S \models\!\!\!| \varphi \wedge \psi \text{ if } S \models\!\!\!| \varphi \text{ or } S \models\!\!\!| \psi$$

And $(\varphi \vee \psi)$ is regarded as an abbreviation for $\sim(\sim\varphi \wedge \sim\psi)$.

In this manner, we can define a nonmonotonic inference relation $\mid\!\sim$ between databases and formulas in $\mathcal{L}(\wedge, \vee, \sim)$, by

$$\Pi \mid\!\sim \varphi \quad \text{iff} \quad S \models \varphi \text{ for all answer sets } S \text{ of } \Pi.$$

For the case of monotonic ('$-$'-free) databases the following characterisation of $\mid\!\sim$ was obtained in Pearce [19].

Proposition 1 *Let Π be a disjunctive database without weak negation '$-$', and φ a sentence from the query language $\mathcal{L}$. Then $\Pi \mid\!\sim \varphi$ iff $\Pi \vdash_N \varphi$.*

Moving to the nonmonotonic case, as pointed out in [20], there is also a simple, logical characterisation of answer sets for XLPs, viz. given a consistent extended logic program Π,

$$S \text{ is an answer set of } \Pi \text{ iff } S = \{L \in Lit : \Pi^S \vdash_N L\}, \tag{3}$$

In the general case where Π is an XDDB, however, (3) no longer holds. Instead we have only the weaker condition:

$$\text{if } S \text{ is an answer set of } \Pi \text{ then } \{L \in Lit : \Pi^S \vdash_N L\} \subseteq S. \tag{4}$$

Equality holds only if every answer set of Π^S is in turn an answer set of Π.

3 The Gödel Translation

I assume the reader is familiar with S4 modal logic as well as the basic ideas of autoepistemic or nonmonotonic modal logic, see, eg. [16, 14, 23]. In the usual manner, modal formulas are build up from propositional variables using the propositional connectives and the necessity operator $\Box$. We use the previous symbols for the connectives, except that in the modal language (material) implication will be denoted by '$\supset$', to emphasise that we are not dealing here with a constructive connective.

The standard (Gödel) embedding of τ of H into modal S4 can be extended to an embedding of the full system N into S4, by specifying extra clauses for the translation of formulas involving strong negation. Restricting attention here to the propositional case, the Gödel translation τ is defined as follows:

$\tau(A) = \Box A$; $\tau(\sim A) = \Box \sim A$, for A an atom;
$\tau(\varphi \wedge \psi) = \tau(\varphi) \wedge \tau(\psi)$
$\tau(\varphi \vee \psi) = \tau(\varphi) \vee \tau(\psi)$
$\tau(\sim(\varphi \wedge \psi)) = \tau(\sim\varphi) \vee \tau(\sim\psi)$
$\tau(\sim(\varphi \vee \psi)) = \tau(\sim\varphi) \wedge \tau(\sim\psi)$
$\tau(\varphi \rightarrow \psi) = \Box(\tau(\varphi) \supset \tau(\psi))$
$\tau(\sim(\varphi \rightarrow \psi)) = \tau(\varphi) \wedge \tau(\sim\psi)$
$\tau(\sim\sim \varphi) = \tau(\varphi)$
$\tau(\sim -\varphi) = \tau(\varphi)$
$\tau(-\varphi) = \Box \sim \tau(\varphi)$.

For a set Φ of propositional N-formulas, let $\tau(\Phi) := \{\tau(\varphi) : \varphi \in \Phi\}$. Then for any formula φ and set of formulas Φ, the following embedding can be established

$$\Phi \vdash_N \varphi \quad \text{iff} \quad \tau(\Phi) \vdash_{S4} \tau(\varphi), \tag{5}$$

where '$\vdash_{S4}$' denotes S4-derivability; ie. in general, for any normal modal system $\mathcal{S}$, we write $T \vdash_{\mathcal{S}} \psi$ to mean that ψ is provable from formulas in T using the $\mathcal{S}$-axioms and the rules of modus ponens and necessitation.[1] If T is a set of modal formulas, we also set

$$Cn_{\mathcal{S}}(T) = \{\varphi : T \vdash_{\mathcal{S}} \varphi\}.$$

Let Π be a consistent extended logic program. Then from (3) and (5) we can conclude that

$$S \subset Lit \text{ is an answer set of } \Pi \text{ iff } S = \{L \in Lit : \tau(\Pi^S) \vdash_{S4} \Box L\}. \tag{6}$$

Similarly, from (5) and (4), for consistent XDDBs we obtain the following:

$$\text{if } S \subset Lit \text{ is an answer set of } \Pi \text{ then } \{L \in Lit : \tau(\Pi^S) \vdash_{S4} \Box L\} \subseteq S. \tag{7}$$

Under the Gödel translation, a weakly negated formula $-\varphi$ is transformed to $\tau(-\varphi) = \Box \sim \tau(\varphi)$. However, since in databases '$-$' only appears directly before a literal, and nowhere else in a database formula, this simplifies to

$$\tau(-L) = \Box \sim \Box L.$$

It is then readily seen that the translation $\tau(\varphi)$ of any formula φ of a disjunctive database Π can be written in the following form:

$$\Box(\Box L_1 \wedge \ldots \wedge \Box L_m \wedge \Box \sim \Box L_{m+1} \wedge \ldots \wedge \Box \sim \Box L_n \supset \Box K_1 \vee \ldots \vee \Box K_k). \tag{8}$$

[1] For the case where Φ is empty, (5) is proved in [26] (with quantifiers included). The general case can be verified by applying the deduction theorem and some simple S4-equivalences. The Gödel embedding for the full system N, including intuitionistic negation, is discussed in [2].

The modal or epistemic interpretation of a database formula is therefore roughly: it is known that, if each of $L_1, \ldots, L_m$ is known and each of $L_{m+1}, \ldots, L_n$ is known not to be known, then K_j is known, for some $j \leq k$. We turn now to the essentials of nonmonotonic modal logic.

For a language of the above kind containing a knowledge or belief operator '$\Box$', the notion of *stable* (belief) set was introduced in [25] and [16]. A belief set S is said to be *stable* iff

(i) S is closed under classical deduction;
(ii) $\varphi \in S \Rightarrow \Box\varphi \in S$;
(iii) $\varphi \notin S \Rightarrow \sim\Box\varphi \in S$.

It is well-known that for any set S of non-modal formulas there is a unique stable belief set T such that the non-modal consequences of T are exactly the logical consequences of S. For any set of $\Box$-free formulas S we denote this unique stable set by $E(S)$. For later use, observe that if S is a consistent set of literals, then for any literal L, $L \in S$ iff $L \in E(S)$.

Let $\mathcal{S}$ be a modal system and T and W sets of formulas in $\mathcal{S}$. As is customary (see eg. [23]), we say that T is an *$\mathcal{S}$-expansion* of W iff

$$T = Cn_{\mathcal{S}}(W \cup \{\sim\Box\varphi : \varphi \notin T\}).$$

For any modal system $\mathcal{S}$ we can consider its nonmonotonic variant equipped with a sceptical inference relation '$\mid\sim_{\mathcal{S}}$' defined by

$$W \mid\sim_{\mathcal{S}} \varphi \quad \text{iff} \quad \varphi \in T, \text{ for all } \mathcal{S}\text{-expansions } T \text{ of } W.$$

We shall deal here with consistent databases and consistent S4-expansions. In particular, this means that for any S4-expansion T of a set of formulas, and any formula φ,

$$(i)\ \varphi \in T \quad \text{iff} \quad \Box\varphi \in T; \quad (ii)\ \varphi \notin T \quad \text{iff} \quad \sim\Box\varphi \in T, \tag{9}$$

so, in particular, for any φ, either $\Box\varphi \in T$ or $\sim\Box\varphi \in T$. It follows that T is a stable set. Notice that any S4-theory, ie. set of formulas closed under S4-derivability, already satisfies (9)(i). Lastly, we state the following as a lemma for later use. It is a special case of a general theorem of Schwarz [23]:

Lemma 1 *If S is a consistent set of literals, then $E(S)$ is the only S4-expansion of S. In other words*

$$E(S) = Cn_{S4}(S \cup \{\sim\Box\varphi : \varphi \notin E(S)\}).$$

4 Answer Sets and S4-Expansions

We are now ready to describe the basic or standard embedding relating the answer sets of XDDBs with S4-expansions, namely

Proposition 2 *Let* Π *be an XDDB and* S *be a consistent set of literals in the language of* Π. *Then the following conditions are equivalent:*
(a) S *is an answer set of* Π;
(b) $E(S)$ *is an S4-expansion of* $\tau(\Pi^S)$;
(c) $E(S)$ *is an S4-expansion of* $\tau(\Pi)$.

This result establishes a correspondence between answer sets and S4-expansions. What does it tell us about the sceptical inference relation, $\mid\sim$, associated with the answer set semantics? In what sense can we derive the answer set consequences of a database by computing, in some manner, the S4-expansions of its modal translation? Clearly, we cannot use the stable sets $E(S)$ directly, since they are classically closed and answer set inference is not. What we can do is to proceed, once again, via the Gödel translation. That is, a Boolean formula φ will be derivable from the database just in case its Gödel translation belongs to each S4-expansion of the translated database. This provides our promised characterisation of answer set inference.

Proposition 3 (Main Theorem) *Let* Π *be a consistent XDDB. For any Boolean formula* $\varphi \in \mathcal{L}(\vee, \wedge, \sim)$, $\Pi \mid\sim \varphi$ *iff* $\tau(\Pi) \mid\sim_{S4} \tau(\varphi)$.

Besides providing a complete characterisation of answer set inference, Proposition 3 yields a monotonic 'lower-bound' for the answer set consequences of any database Π. Since every S4-expansion of $\tau(\Pi)$ contains $\tau(\Pi)$ and is closed under deduction in S4, clearly, for any $\varphi \in \mathcal{L}$, $\tau(\Pi) \vdash_{S4} \tau(\varphi) \Rightarrow \tau(\Pi) \mid\sim_{S4} \tau(\varphi)$. Applying the Gödel embedding (5) once again, it follows that for any φ

$$\Pi \vdash_N \varphi \Rightarrow \Pi \mid\sim \varphi. \tag{10}$$

Using (10) to compute a monotonic 'lower-bound' for answer set inference has the advantage that we do not need to translate formulas into S4. Instead we can use the ordinary tableau or sequent calculus for N, suitably trimmed to take account of the restricted syntax of disjunctive databases. Then if a Boolean formula is provable from Π in the restricted N-calculus, it is certainly a consequence of Π in the answer set semantics.

4.1 Related Modal Embeddings

Since S4 is not the only modal logic into which N can be embedded it would be surprising if we could not extend and strengthen Proposition 2 so as to obtain additional correspondences between answer sets and nonmonotonic modal logics. As we mentioned at the beginning, alternative proposals have already been made in [13] and [12].

Marek & Truszczynski [13] consider a normal modal logic, SW5, that lies between S4 and S5 and possesses simple frames. It is characterised by S4 together with the following axiom, called W5:

$$\sim\Box \sim\Box\varphi \supset (\varphi \supset \Box\varphi).$$

[13] employs a slightly simplified modal translation of database formulas. Any such formula of the form (2) is translated into the modal formula:

$$\Box L_1 \wedge \ldots \wedge \Box L_m \wedge \Box \sim \Box L_{m+1} \wedge \ldots \wedge \Box \sim \Box L_n \supset \Box K_1 \vee \ldots \vee \Box K_k. \quad (11)$$

Let $\sigma(\varphi)$ denote the above translation, for any $\varphi \in \Pi$. Then clearly $\tau(\varphi) = \Box\sigma(\varphi)$. Consequently, for any $\varphi \in \Pi$, we have

$$\tau(\varphi) \vdash_{S4} \sigma(\varphi); \quad \sigma(\varphi) \vdash_{S4} \tau(\varphi).$$

Moreover, in [13] a correspondence is established between the answer sets of a database Π and the SW5-expansions of $\sigma(\Pi)$. This correspondence can also be deduced from Proposition 2. First, notice that, since S4 $\subseteq$ SW5 $\subseteq$ S5, a general result of [16] implies that for any sets T, W of modal formulas, if T is an S4-expansion of W then it is an SW5-expansion of W. Consequently, from Proposition 2 we can conclude that if S is an answer set of Π, then $E(S)$ is an SW5-expansion of $\tau(\Pi^S)$ and of $\tau(\Pi)$. Since any SW5-theory contains the formulas of $\tau(\Pi)$ if and only if it contains those of $\sigma(\Pi)$, $E(S)$ is also then an SW5-expansion of $\sigma(\Pi)$. The converse can be proved using essentially the argument for Proposition 2 ($(c) \Rightarrow (a)$).

Just as one can strengthen the modal host logic into which constructive systems can be interpreted, so can one vary to some extent the nature of the interpretation. Under the standard Gödel translation, as well as some others like the Tarski interpretation of H in S4 ([15]), the host logic has, like S4 and SW5, reflexive frames. But in modal logic there are well-known techniques for interpreting such logics, in their turn, into modal systems possessing non-reflexive or even irreflexive frames (cf. some examples presented in [22, Ch. 5]). An appropriate combination of the two interpretations may then produce an embedding of a constructive system into a non-reflexive modal system. Roughly-speaking the method consists in mimicking reflexivity by replacing every boxed atomic formula, $\Box A$, by the conjunction $A \wedge \Box A$, and iterating this construction for arbitrary formulas of the language. The technique is used by Schwarz [24] to obtain a mutual embedding between the nonmonotonic versions of SW5 and KD45. The latter is, of course, nonreflexive and is well-known as being the nonmonotonic modal logic corresponding to Moore's autoepistemic logic. Accordingly, nonmonotonic SW5 amounts to a reflexive version of autoepistemic logic, appropriately termed *reflexive autoepistemic logic* in [13]. Further, by composing the interpretation σ with the above transformation $\Box A \rightsquigarrow A \wedge \Box A$, one obtains, as [13] shows, an embedding of the answer set semantics into nonmonotonic KD45, hence into autoepistemic logic. This embedding was independently obtained (directly) by Lifschitz & Schwarz [12] and by Chen [4]. The resulting translation is syntactically more complex than σ or τ and is not especially transparent. Its only *raison d' être* seems to be that it succeeds in relating the semantics of databases to autoepistemic logic. Moreover, it can be analysed as the composition of two quite separate interpretations, both of which are historically well-founded and well-understood. Unless one gives special value to

working with nonreflexive frames, or employing autoepistemic logic in its original formulation, there seems to be no advantage gained by using the hybrid translation. Moreover, if one interprets possible-world frames according to the standard reading of epistemic logic – that the accessibility relation on worlds represents epistemic or doxastic alternatives – then reflexivity, or the property that every world is an epistemic alternative to itself, is a very natural condition to impose.

5 Concluding Remarks

We began with the task of obtaining a logical, declarative interpretation of extended disjunctive databases or logic programs equipped with an answer set semantics; and thereby to characterise the (sceptical) nonmonotonic inference relation associated with this semantics. Since database clauses are usually represented as rules rather than logical formulas, their connectives regarded as being in some sense 'nonstandard', and their semantics based on a fixpoint characterisation involving the Gelfond-Lifschitz *reduction* of a database, it is not immediately evident to what extent the ideal of 'programming in logic' still applies to such a system. Fortunately, our problem can be neatly broken-down into two subtasks: (i) isolate a monotonic subsystem and provide a logical characterisation of its inference patterns; (ii) find a suitable nonmonotonic extension of the latter, which will characterise the system as a whole.

The first subtask was already solved in [19] by noting that the clauses of *not*-free databases can be identified with formulas of constructive logic with strong negation N, whose inference relation then precisely coincides with that of (sceptical) answer set inference. It follows that the nonmonotonic inference relation, $\mid\!\sim$, associated with answer sets for the full system of extended disjunctive databases can be regarded as an extension of the inference relation, $\vdash_N$, of N. I have discussed elsewhere ([21]) some of the general properties of $\mid\!\sim$, regarded as a *supraconstructive* inference relation. Since constructive logic subsumes intuitionistic logic, in the sense that Heyting's negation '$-$' is definable in N in such a way that all theorems of H become theorems of N, it is rather natural to stick with the logical vocabulary of N and to identify the weak negation '*not*' of database formulas with the intuitionistic negation '$-$' of N. Clearly, we cannot then employ the monotonic inference relation of N to interpret $\mid\!\sim$, but require instead a suitable nonmonotonic extension of $\vdash_N$.

At this juncture there appear to be two ways to proceed. One strategy would be to design a suitable nonmonotonic logic based directly on N, say by introducing a modal consistency or non-provability operator M, and giving rules for its use. Designing such a system remains, however, a task for future research. Instead we adopted here a simpler, alternative strategy. Since N is faithfully interpretable in modal S4, we used S4 directly in its nonmonotonic version. The advantages are that the system is well-understood, and, moreover, it appears to be the only known example of a modal logic whose nonmonotonic variant is less complex than its monotonic ancestor.

Using the Gödel translation τ of all of the connectives of N (including both weak and strong negation), one obtains a correspondence between answer sets of a database and the S4-expansions of its translation. The nature of this correspondence is such that any Boolean formula is a nonmonotonic consequence of the database (under the answer set semantics) if and only if its Gödel translation is a consequence in nonmonotonic S4 of the translated database. Moreover, as we saw, it is straightforward to extend this embedding to other modal host logics. Any method for 'computing' the intersection of all S4-expansions can therefore be applied to determine the answer set consequences of a logic program or disjunctive database. However, as far as characterising a monotonic lower-bound is concerned, it may be simpler to work directly in sequent, natural deduction or tableau calculi for N.

Acknowledgements

This is a preliminary version of a paper extending research reported by the author at the 4th International Workshop on Extensions of Logic Programming in St Andrews and at the 2nd International Workshop on Logic Programming and Nonmonotonic Reasoning in Lisbon, 1993. The present version has benefitted greatly from the presentations of [12] and [13] in Lisbon and I am especially grateful to Wiktor Marek for pointing out to me the close connection between our two modal translations of database formulas, and for the interest he showed in the results of [20]. I am also grateful to Marcus Kracht for valuable discussions and advice. The work reported here was carried out within the research project *Systems of Logic as a Theoretical Foundation for Knowledge and Information Processing* supported by the Freie Universität Berlin. An extended version of this paper, including proofs of the new results, is being submitted for publication.

References

[1] Akama, S, Constructive Predicate Logic with Strong Negation and Model Theory, *Notre Dame J Formal Logic* 29 (1988), 18–27.

[2] Akama, S, On the Proof Method for Constructive Falsity, *Zeitschr. f. math. Logik und Grundlagen d. Math.* 34 (1988), 385–392.

[3] Almukdad, A & Nelson, D, Constructible Falsity and Inexact Predicates, *JSL* 49 (1984), 231–233.

[4] Chen, J, Minimal Knowledge + Negation as Failure = Only Knowing (Sometimes), in [18].

[5] van Dalen, D, Intuitionistic Logic, in D Gabbay, & F Guenthner (eds), *Handbook of Philosophical Logic, Vol. III*, Kluwer, Dordrecht, 1986. ag, 1982, 260–273.

[6] Gelfond, M, On Stratified Autoepistemic Theories, in Proc AAAI-87, 1987, 207–211.

[7] Gelfond, M, Logic Programming and Reasoning with Incomplete Information, Manuscript, 1992; to appear in *Annals of Mathematics and Artificial Intelligence.*

[8] Gelfond, M & Lifschitz, V, The Stable Model Semantics for Logic Programs, in R Kowalski & K Bowen (eds), *Proc ICLP-88*, 1070–1080.

[9] Gelfond, M & Lifschitz, V, Logic Programs with Classical Negation, in D Warren & P Szeredi (eds), *Proc ICLP-90*, MIT Press, 1990, 579–597.

[10] Gelfond, M & Lifschitz, V, Classical Negation in Logic Programs and Disjunctive Databases, *New Generation Computing* 9 (1991), 365–385.

[11] Gurevich, Y, Intuitionistic Logic with Strong Negation, *Studia Logica* 36 (1977), 49–59.

[12] Lifschitz, V, & Schwarz, G, Extended Logic Programs as Autoepistemic Theories, in [18].

[13] Marek, V & Truszczynski, M, Reflexive Autoepistemic Logic and Logic Programming, in [18].

[14] McDermott, D, & Doyle, J, Non-monotonic Logic I, *Artificial Intelligence* 13 (1980), 41–72.

[15] McKinsey, J, & Tarski, A, Some Theorems about the Sentential Calculi of Lewis and Heyting, *JSL* 13 (1948), 1–14.

[16] Moore, R C, Semantical Considerations on Non-Monotonic Logic, *Artificial Intelligence* 25 (1985), 75–94.

[17] Nelson, D, Constructible Falsity, *JSL* 14 (1949), 16–26.

[18] Pereira, L M, & Nerode, A (eds), *Logic Programming and Non-monotonic Reasoning*, MIT Press, 1993.

[19] Pearce, D, Answer Sets and Constructive Logic. Part I: Monotonic Databases, to appear in Fuhrmann, A, & Rott, H, (eds), *Logic, Action and Change*, de Gruyter, Berlin, to appear.

[20] Pearce, D, Answer Sets and Constructive Logic, II: Extended Logic Programs and Related Nonmonotonic Formalisms, in [18]

[21] Pearce, D, Remarks on Nonmonotonicity and Supraclassicality, 1993, to appear as a Technical Report, FU Berlin, 1994.

[22] Rautenberg, W, *Klassische und Nichtklassische Aussagenlogik*, Vieweg, Wiesbaden, 1979.

[23] Shvarts, G, Autoepistemic Modal Logics, in R Parikh (ed), *Proceedings TARK-90*, Morgan Kaufmann, 1990.

[24] Schwarz, G, Autoepistemic Logic of Knowledge, in A Nerode *et al* (eds), *Logic Programming and Non-Monotonic Reasoning*, MIT Press, 1991, 3–20.

[25] Stalnaker, R, A Note on Non-Monotonic Modal Logic, Manuscript, Cornell University, 1980.

[26] Thomason, R H, A Semantical Study of Constructible Falsity, *Zeit. f. math. Logik und Grundlagen der Mathematik* 15 (1969), 247–257.

[27] Vorob'ev, N N, A Constructive Propositional Calculus with Strong Negation (in Russian), *Doklady Akademii Nauk SSR* 85 (1952), 465–468.

[28] The Problem of Deducibility in Constructive Propositional Calculus with Strong Negation (in Russian), *Doklady Akademii Nauk SSR* 85 (1952), 689–692.