Answer Set Programming and S4

Mauricio Osorio and Juan Antonio Navarro

Universidad de las Américas, CENTIA
Sta. Catarina Mártir, Cholula, Puebla. 72820 México
josorio@mail.udlap.mx

Abstract. We present a proposal on how to develop nonmonotonic reasoning based on the modal logic S4. As a consequence of our results we show how to present the well known semantics of answer sets, under a restricted fragment of modal formulas, using this approach. Moreover, by considering the full set of modal formulas we obtain an interesting generalization of answer sets with modal connectives. We show, with several examples, possible applications of this proposed inference system and how the syntax of modal formulas can clearly describe notions such as negation as failure.

It is also possible to replace the modal logic S4 with any other modal logic to obtain similar nonmonotonic systems. We even consider the use of multimodal logics in order to model a scenario where several agents can reason about the knowledge and beliefs of each other. Our results clearly state interesting links between answer set programming, modal logics and multi-agent systems which might bring research of these areas together.

1 Introduction

Since its introduction in 1988 by Gelfond and Lifschitz the stable model semantics has been recognized as a relevant and novel contribution to the communities of nonmonotonic reasoning and logic programming. Several research groups in Europe, Japan and U.S.A. are developing theory and applications related to the stable model semantics.

The paradigm that originated from the stable model semantics is known today as Answer Set Programming (ASP). The basic idea of ASP is to provide a formal system that assigns to each theory some desirable models or extensions called *answer sets*. The development efficient implementations of answer sets finders also allowed the production of several applications that range from planning, solving combinatorial problems, verification, logical agent systems and product configuration.

On the other hand modal logic was developed as a concequence of the study of notions such as "necessary" and "possible". Extending the syntax of logic formulas with new unary connectives $\Box$ and $\Diamond$, and giving an adequate semantical meaning to them, it is possible to define a formal system to model notions like knowledge, tense, obligation, and many others. Modal formulas usually have a very natural reading close to their intended intuitive meaning. That is one of the reasons of why this kind of logics have been used to provide foundations to several applications in knowledge representation, multi-agent systems, etc.

In this paper, we propose to use the modal logic S4 in a general framework to model nonmonotonic reasoning. This result follows from a characterization of answer sets in terms of intuitionistic logic we provided recently. The characterization, as well as several properties and consequences, are presented in [7–9]. Other results about modal logics and semantics have been presented in [6].

Moreover, we prove that the ASP approach is embedded in our proposed S4 nonmonotonic semantics. This is not really a surprise since, by virtue of the characterization of answer sets, intuitionistic logic can be embeded into modal logic S4 (thanks to a well known Gödel's translation).

We propose the following interpretation to our system: Consider a logic agent with some modal theory as its base knoeledge. The agent could use the logic S4, a logic of knowledge, in order to do inference and produce new knowledge. However, we would also like our agent to be able to do nonmonotic reasoning.

Informally speaking we will allow our agent to suppose some *simple acceptable knowledge* in order to make more inference. This simple acceptable knowledge consists of formulas of the form $\Box\Diamond F$, with F a formula with only unary connectives, that literally states that the agent knows that some fact is possible. If building such extensions the agent is able to justify his assumptions (to prove all formulas F) and obtain some sort of *complete* explanation to his base knowledge then the agent can use this extension to do more inference.

The fact that nonmonotonic reasoning can be done via S4 has been already shown in [13]. Our approach is different from others since it tries to find relations between the nonmonotonic semantics of answer sets and, in particular, the modal logic S4. The study of such relations even allows to present a new characterization of answer sets. This characterization generalizes results by Pearce in [10], where he only considers the class of disjunctive programs.

Our paper is structured as follows: In Section 2 we present the general background of our paper, we briefly introduce the syntax of our programs and some notation. In Section 3 we present our framework for doing nonmonotonic reasoning using the logic S4. In Section 4 we introduce a previous result that relates answer sets with intuitionistic logic and, in Section 5, we establish the relations with respect to the logic S4. Finally we present in Section 6 some ideas on how to develop an ASP approach for multi-agent systems formulating an example with two different agents. We terminate presenting some conclusions in Section 7.

2 Background

In this section we briefly introduce some basic concepts and definitions that will be used along this paper. We introduce the language of propositional modal logic and present a proof system for the modal logic S4.

2.1 Propositional Modal Logic

We use the set of propositional modal formulas in order to describe rules and information within logic programs. Formally we consider a language built from an alphabet containing: a denumerable set $\mathcal{L}$ of elements called *atoms* or *atomic formulas*; the binary connectives $\wedge$, $\vee$ and $\rightarrow$ to denote conjunction, disjunction and implication respectively; the unary connective $\Box$ as a *knowledge operator*; the 0-ary connective $\bot$ to denote falsity; and auxiliary symbols (,).

Formulas can be constructed as usual in logic. The negation $\neg F$ can be introduced as an abbreviation of the formula $F \rightarrow \bot$, the *belief operator* $\Diamond F$ to abbreviate $\neg\Box\neg F$ and, similarly, the truth symbol $\top$ that stands for $\neg\bot$. We also can write, as usual, $F \leftrightarrow G$ to denote the formula $(F \rightarrow G) \wedge (G \rightarrow F)$. Finally, the formula $G \leftarrow F$ is just another way of writing $F \rightarrow G$.

A *modal theory*, or *modal program*, is a set of modal formulas, we restrict our attention however to finite theories. For a given theory T its *signature*, denoted $\mathcal{L}_T$, is the set of atoms that occur in the theory T. Observe that, since we consider finite theories, their signatures are also finite. Given a theory T we also define the negated set $\neg T = \{\neg F \mid F \in T\}$ and the knowledge set $\Box T = \{\Box F \mid F \in T\}$.

A *literal* is either a formula of the form a (positive literal) or $\neg a$ (negative literal) where a is an atom. Given a theory T we use $Lit_T = \mathcal{L}_T \cup \neg\mathcal{L}_T$ to deonte the set of all literals that are relevant to T. If, for instance, we have the theory $T = \{\neg a \rightarrow b\}$ then $Lit_T = \{a, \neg a, b, \neg b\}$.

2.2 Modal Logic S4

Modal logic was originally conceived as the logic of necessary and possible. The logic S4 can be defined as the Hilbert type proof system that contains the following axiom schemes:

1. $(F \rightarrow (G \rightarrow H)) \rightarrow ((F \rightarrow G) \rightarrow (F \rightarrow H))$
2. $(F \rightarrow (G \rightarrow F)$
3. $\neg\neg F \rightarrow F$
4. $\Box(F \rightarrow G) \rightarrow (\Box F \rightarrow \Box G)$
5. $\Box F \rightarrow \Box\Box F$
6. $\Box F \rightarrow F$

and is closed under the rule of Necessitation (from F we can derive $\Box F$) as well as Modus Ponens (from F and $F \rightarrow G$ we can derive G). The behaviour of other connectives follows from their usual definition in classical logic.

We use the standard notation $\vdash F$ to denote that F is a provable formula in the logic S4. If T is a theory we understand the symbol $T \vdash F$ to mean that $\vdash F_1 \wedge \cdots \wedge F_n \rightarrow F$ for some F_i contained in T. Similarly, given a theory U, we use the symbol $T \vdash U$ to denote $T \vdash F$ for every $F \in U$. A theory T is said to be consistent, with respect to the logic S4, if it is not the case that $T \vdash \bot$. We use the notation $T \Vdash U$ to stand for the phrase: T is consistent and $T \vdash U$.

3 Non-monotonic Reasoning in S4

In this section, we introduce our proposal on how to do nonmonotonic inference in S4. Our main concept is the notion of weakly complete and consistent extension of a given theory. The idea of our approach is as follows: *"If we cannot* *derive a formula F by a standard inference in S4 from a theory T, we could try to derive the formula F by a 'suitable' extension of the theory T."*

The first basic requirement of this extended theory is to be consistent. Second, the extra formulas that we include should be a sort of 'weak' assumptions. Any formula of the form $\Box\Diamond F$ (where F is any formula having only 1-place connectives) is considered a 'weak' assumption. Such formula just says that it is known that is possible something. As noted before, we borrow such formula if it helps us to obtain a consistent explanation of the world.

Definition 1. *Let P be any modal theory and $M \subseteq Lit_P$. The* modal closure *of M is defined as $\overline{M} = \neg(Lit_P \setminus M) \cup \Box M$ and the elements in $\overline{M}$ are known as* simple acceptable knowledge. *We define $\Box M$ to be an* S4-answer set *of P if and only if $P \cup \Box\Diamond\overline{M} \Vdash_{S4} \Box M$. Moreover $P \cup \Box\Diamond\overline{M}$ is called a* weakly complete and consistent extension *of P.*

Consider the program $P = \Box(\neg\Box a \rightarrow b)$ with $Lit_P = \{a, \neg a, b, \neg b\}$. Observe that $\{\Box b\}$ is an S4-answer since $P \cup \Box\Diamond\{\neg\neg a, \neg a, \neg\neg b, \Box b\}$ is consistent and proves, under S4, the formula $\Box b$. This program has no more answer sets. We could also consider the following more interesting example:

1. Juan is mexican.
2. Mary is american.
3. Pablo is mexican and not catholic.
4. It is known that normally mexicans are catholic.

We can encode this problem using the following program:

$\Box(mexican(juan))$.
$\Box(american(mary))$.
$\Box(mexican(pablo) \wedge \neg catholic(pablo))$.
$\Box((mexican(X) \wedge \neg\Box\neg catholic(X)) \rightarrow catholic(X))$.

Note that the sentence "It is known that normally mexicans are catholic" is encoded by the last rule, that says: It is known that 'if it is not known that X is not catholic' and 'X is mexican' then 'X is catholic'. Of course, we also need (not just S4) our extended nonmonotonic S4 to make this example work as we immediately explain. The unique s4-answer set of this program is:

$$\Box\{\neg catholic(pablo), mexican(pablo), mexican(juan),\\ catholic(juan), american(mary)\}$$

So, we know that Pablo is not catholic. We also know that Juan is catholic. However, this knowledge comes from nonmonotonic reasoning. This kind of knowledge derived not by the regular inference procedure of S4 can be considered as a kind of "weak knowledge" or a "strong belief" so to speak. The s4-answer set does not say if Mary were catholic or not since the program does not provide any suggestion about this fact.

Why modal logic S4? The S4 and S5 systems are perhaps the two most well known systems to represent the notions of knowledge and possibility. The theorem that we present in the next section however fails if we use S5 instead of S4. The problem is that S5 has no "irreducible iterated modalities". In S5, $\Box F$ is equivalent to $\Diamond\Box F$. We need in our approach to distinguish both formulas in order to gain expressibility. There are however many logics between S4 and S5 that behave nicely with respect to our approach.

4 Expressing Answer Sets

The answer set semantics is a popular semantic operator for logic programs. One of the main features of answer sets are is the introduction of negation as failure[1] which is extremely useful to model notions such as nonmonotonic reasoning, default knowledge and inertial rules.

The definition of answer sets is not required for the purposes of this paper, the reader is referred to [5] for more details. It is just importat to mention that answer sets are defined for augmented logic programs, a class of propositional logic programs (without modal connectives), and incorporates an additional *classical negation* connective, denoted $\sim$, that can only be used before atomic ocurrences.

4.1 Logical Foundations of A-Prolog

The characterization of answer sets in terms of intermediate logics is an importat result that provides a solid logical foundation to this paradigm. Pearce initiated this line of research using intuitionistic extensions, composed by negated atoms, to characterize answer sets for disjunctive logic programs [12]. His characterization, however, was not intended to characterize answer sets for other classes of programs. In fact the characterization is not able to capture the answer sets of logic programs containing negation in the head [8].

Pearce also developed another approach using extensions of theories based on the logic HT, and showed that they are equivalent to the *equilibrium logic* also developed by himself [11, 12]. In a more recent paper [4], together with Lifschitz and Valverde, Pearce was able to show that equilibrium models actually characterize answer sets for the class of augmented logic programs.

Following the original idea from Pearce we were able to show, however, that a characterization of answer sets for augmented programs in terms of intuitionistic logic is also possible. We do consider extensions with negated atoms, as Pearce did, but also allow double negated atoms ($\neg\neg a$) in our intuitionistic extensions. In [8] the following theorem was stated and proved.

Theorem 1. *Let P be an augmented program and let $M \subseteq \mathcal{L}_P$. M is an answer set of P if and only if $P \cup \neg(\mathcal{L}_P \setminus M) \cup \neg\neg M \Vdash_{\mathrm{I}} M$.*

Observe that this theorem asumes that augmented programs do not allow classical negation. This note is important since classical negation has no defined

[1] The negation symbol we use ($\neg$) will play the role of the negation as failure in logic programs. The authors in [5] use, however, the symbol *not* instead.

meaning under intuitionistic logic. The assumption is justified, however, since classical negation can be easily simulated under a proper renaming of atoms as will be explained in Section 4.2.

This characterization also suggests a natural way to define a notion of answer sets for any propositional theory T. This was also proposed in [8] and used later to develop the *safe belief semantics* in [9].

Definition 2. *Let P be any propositional theory and let $M \subseteq \mathcal{L}_P$. We define M to be an answer set of P if and only if $P \cup \neg(\mathcal{L}_P \setminus M) \cup \neg\neg M \Vdash_{\mathrm{I}} M$.*

We have also shown that the answer set semantics is invariant under any si-logic and generalized this approach to include extensions of the form $\neg\neg F$, with F any formula, providing a more general framework to study and define semantics, see [9]. These results are (in our point of view) good evidence of the well-behaviour of the answer set semantics in logical terms.

4.2 Restoring Classical Negation

We will now show how to restore the use of classical negation in our logic programs. Recall that $\sim$ is only allowed in front of atomic formulas, intuitively we can think of the formula $\sim a$ as an atom with a convenient name so that an answer set finder can discard models where both a and $\sim a$ appear.

Formulas of the form a and $\sim a$ are referred as $\sim$*literals*, and we say that a set of $\sim$literals M is consistent if it is not the case that both a and $\sim a$ are contained in M for some atom a. We will also use the terms *asp-formulas*, *asp-theories* and *asp-programs* to denote entities that allow the use of classical negation. We also define, if M is a set of atoms, $\sim M = \{\sim a \mid a \in M\}$. Moreover, for an asp-program P its *extended signature* will be $\mathcal{L}_P^{\sim} = \mathcal{L}_P \cup \sim\mathcal{L}_P$.

Definition 3. *Given a signature $\mathcal{L}$, let $\mathcal{L}'$ be another signature with the same cardinality as $\mathcal{L}$ and with $\mathcal{L} \cap \mathcal{L}' = \emptyset$. Also let $f\colon \mathcal{L} \to \mathcal{L}'$ be a biyective function between the two signatures. We define the mapping $+$ from asp-formulas over the signature $\mathcal{L}$ to formulas over $\mathcal{L} \cup \mathcal{L}'$ recursively as follows:*

1. $(\bot)^+ = \bot$.
2. *for any atom a let $(a)^+ = a$ and $(\sim a)^+ = f(a)$.*
3. *for any pair of formulas F, G let $(F \odot G)^+ = F^+ \odot G^+$ where $\odot \in \{\wedge, \vee, \to\}$.*

The translation $+$ can also be defined over theories as usual, $T^+ = \{F^+ \mid F \in T\}$.

Lemma 1. *Let P be an augmented asp-program and let $M \subseteq \mathcal{L}_P^{\sim}$ be a consistent set of $\sim$literals. Let $+$ be a mapping defined with a proper set $\mathcal{L}'_P$ and a function $f\colon \mathcal{L}_P \to \mathcal{L}'_P$. M is an answer set of P iff M^+ is an answer set of P^+.*

Proof. Follows as a direct generalization of Proposition 2 in [1].

Theorem 2. *Let P be an augmented asp-program, and let M be a consistent set of $\sim$literals. M is an answer set of P iff $P^+ \cup \neg(\mathcal{L}_{P^+} \setminus M^+) \cup \neg\neg M^+ \Vdash_{\mathrm{I}} M^+$.*

Proof. Follows immediately by Theorem 1 and previous lemma.

5 Characterization of Answer Sets using S4

The purpose of the following translation, given in [2], would be to provide a meaning to any propositional theory, including classical negation with a broader role, and not only for the class of augmented programs. We can consider, for instance, the use of the classical negation connective in front of any formula and not only for atoms. We will see that propositional modal formulas are expressive enough to model the two kinds of negations in asp-programs.

Definition 4. *The translation* $^\circ$ *of asp-formulas to modal formulas is defined recursively as follows:*

$(a)^\circ = \Box a,$ *for atomic* a	$(a)^* = \Box \neg a,$ *for atomic* a
$(F \vee G)^\circ = F^\circ \vee G^\circ$	$(F \vee G)^* = F^* \wedge G^*$
$(F \wedge G)^\circ = F^\circ \wedge G^\circ$	$(F \wedge G)^* = F^* \vee G^*$
$(F \to G)^\circ = \Box(F^\circ \to G^\circ)$	$(F \to G)^* = F^\circ \wedge G^*$
$(\neg F)^\circ = \Box \neg F^\circ$	$(\neg F)^* = F^\circ$
$(\sim F)^\circ = F^*$	$(\sim F)^* = F^\circ$

The definition is also extended to sets of asp-formulas, $T^\circ = \{F^\circ \mid F \in T\}$.

Observe that our translation behaves just like the Gödel's translation of intuitionistic logic into S4 for programs without classical negation.

Due to the well known Gödel embedding of intuitionistic logic in S4 and Theorem 2, we can express answer sets for augmented programs, where the role of classical negation is "passive" (because it is only applied to atoms).

Theorem 3. *Let* P *be an augmented asp-program, and let* M *be a consistent set of* $\sim$*literals. The set* M *is an answer set of the logic program* P *if and only if* $(P^+ \cup \neg(\mathcal{L}_{P^+} \setminus M^+) \cup \neg\neg M^+)^\circ \Vdash_{S4} {M^+}^\circ$.

In the rest of this section we discuss an alternative representation of answer sets in S4 aiming our attention to construct the basis of modeling an active role of classical negation.

Definition 5. *Let* $\mathcal{L}$ *and* $\mathcal{L}_1$ *two disjoint signatures and* f *a bijective function from* $\mathcal{L}$ *to* $\mathcal{L}_1$*. We define a mapping* $-$ *from modal formulas with signature* $\mathcal{L} \cup \mathcal{L}_1$ *to formulas in* $\mathcal{L}$ *recursively as follows:*

1. $(\bot)^- = \bot$.
2. *for any atom* a *let* $(a)^- = a$ *if* $a \in \mathcal{L}$ *and* $(a)^- = \neg f^{-1}(a)$ *otherwise.*
3. *for any pair of formulas* F, G *let* $(F \odot G)^- = F^- \odot G^-$ *where* $\odot \in \{\wedge, \vee, \to\}$.
4. *for any formula* F *let* $(\Box F)^- = \Box F^-$.

We also extend our translation over theories: $T^- = \{F^- \mid F \in T\}$.

Lemma 2. *If* P *is an augmented asp-program, defined over* $\mathcal{L}$*, then* ${P^{+}}^{\circ -} = P^\circ$.

Proof. Without lost of generality it suffices to consider a singleton program. Then, it suffices to apply a direct induction on the size of the formula.

Theorem 4. *Let P be an augmented asp-program, and let M be a consistent set of $\sim$literals. Then M is an answer set of P iff M° is a* S4*-answer set of P°.*

Proof. Let $\mathcal{L}'_P$ be a new signature with the same cardinality as $\mathcal{L}_P$ and such that $\mathcal{L}'_P \cap \mathcal{L}_P = \emptyset$. Let $f \colon \mathcal{L}_P \to \mathcal{L}'_P$ be a bijective function. We have then that a consistent set of $\sim$ literals M is an answer set of P:

iff $P^+ \cup \neg(\mathcal{L}_{P^+} \setminus M^+) \cup \neg\neg M^+ \Vdash_{\mathrm{I}} M^+$, by Theorem 2.
iff $(P \cup \neg(\mathcal{L}_P^{\sim} \setminus M) \cup \neg\neg M)^+ \Vdash_{\mathrm{I}} M^+$, by the definition of the mapping $+$.
iff $(P \cup \neg(\mathcal{L}_P^{\sim} \setminus M) \cup \neg\neg M)^{+^\circ} \Vdash_{\mathrm{S4}} M^{+^\circ}$, by Gödel's embedding of I into S4.
iff $(P \cup \neg(\mathcal{L}_P^{\sim} \setminus M) \cup \neg\neg M)^{+^{\circ^-}} \Vdash_{\mathrm{S4}} M^{+^{\circ^-}}$, by a proof by induction on the length of the S4 proof and since M is a consistent set of $\sim$literals (the very restricted class of formulas added as an extension is also required).
iff $(P \cup \neg(\mathcal{L}_P^{\sim} \setminus M) \cup \neg\neg M)^\circ \Vdash_{\mathrm{S4}} M^\circ$, by Lemma 2.
iff $P^\circ \cup \Box\Diamond(\neg(Lit_{P^\circ} \setminus M^*) \cup M^\circ) \Vdash_{\mathrm{S4}} M^\circ$, by set theory and definition of $\circ$, where M^* is obtained from M replacing $\sim$ with $\neg$. Note that $M^\circ = \Box M^*$, also that $(\neg S)^\circ = \Box\Diamond\neg S^*$ and $(\neg\neg S)^\circ = \Box\Diamond\Box S^*$ for any set of $\sim$literals S.
iff M° is an s4-answer set of P°, by the definition of S4-answer sets.

It is also possible to show, thanks to the invariance of the answer set semantics with respect to intermediate logics proved in [9], that any modal logic strictly weaker than S4.3 can be used to characterize answer sets.

Theorem 5. *Let P be an augmented asp-program, and let M be a consistent set of $\sim$literals. Then M is an answer set of P iff M° is a* X*-answer set of P°. Where X is any logic between S4 and S4.3.*

Proof. Follows from Theorem 4 and results in [9].

The transformations proposed and studied in [2] exhibit, in particular, an embedding of the logic of Nelson N into S4. Recall that the logic of Nelson already considers two kind of negations which correspond to the $\neg$ and $\sim$ we introduced here. We believe that, based on the results presented on this dissertation thesis, the proofs of theorems presented in this section could be simplified. It would also make the introduction of $\sim$ less artificial.

6 ASP for multimodal logic

ASP for multimodal logic requires to generalize Definition 1. Instead of that we only present an example and leave a formal presentation for a future paper. We will consider *"The Wise Man Puzzle"* as an example to show how to integrate nonmonotonic reasoning when the knowledge and beliefs of several agents play an important role. This puzzle is typically stated as follows [3]:

> *A king wishes to determine which of his three wise men is the wisest. He arranges them in a circle so that they can see and hear each other and tells them that he will put a white or black spot on each of their foreheads but that at least one spot will be white. In fact all three are white. He offers his favor to the one who first tells him the color of his spot. After a while, the wisest announces that his spot is white. How does he Know?*

The solution is based, of course, on the ability of the agents involved to reason about knowledge and beliefs of other agents, information that can be observed and the rules stated by the king at the begining of the game. We will model a shorter and simpler version in which only two wise man participate. We write w_1 (w_2) to denote that the first (second) wise man has a white spot on his forehead. We can state a set of assumptions P as follows:

$$\begin{array}{ll}
\Box_1(w_1 \vee w_2), \Box_2(w_1 \vee w_2), & (1) \\
\Box_1\Box_2(w_1 \vee w_2), \Box_2\Box_1(w_1 \vee w_2), & \\
\Box_1(w_1 \to \Box_2 w_1), \Box_2(w_2 \to \Box_1 w_2), & (2) \\
\Box_1(\neg w_1 \to \Box_2 \neg w_1), \Box_2(\neg w_2 \to \Box_1 \neg w_2), & \\
\Box_2\Diamond_2\Diamond_1 w_1 \to \Box_2\Diamond_1 w_1, & (3) \\
\Box_2\Diamond_2\Diamond_1 \neg w_1 \to \Box_2\Diamond_1 \neg w_1, &
\end{array}$$

The group of rules under block (1) state the fact that each wise man knows the king's announcement: *"at least one spot will be white"*, and they also know each other knows this information. The rules on block (2) state that they know that they can see each other. Finally the pair of rules under (3) are the nonmonotonic part of our program. Those rules will make the second wise man assume, if it makes sence, that the first wise man believes w_1 (or $\neg w_1$).

Note that $P \not\vdash_{\mathrm{S4}} \Box_2 w_2$ since, in principle, the second wise man can not know *for sure*, using only the information in P, the fact that he has a white spot. He cloud try, however, to do nonmonotonic inference extending his theory with simple acceptable knowledge. Adding $\Box_2\Diamond_2\Diamond_1\neg w_1$ to P allows him, in fact, to prove that $\Box_2 w_2$. If we take the set $M = \Box_2\Diamond_2 \{\Box_2 w_1, \Box_2 w_2, \Diamond_1 w_1, \Diamond_1 \neg w_1, \Box_1 w_2\}$ as a set of simple acceptable knowledge, it turns out that $P \cup M$ is consistent and proves the facts $\{\Box_2 w_1, \Box_2 w_2, \Box_1 w_2\}$. A reasonable definition of ASP for multimodal logic should recognise this set as a valid answer set.

As an important observation notice that we did not have to explicitly include $w_1 \wedge w_2$ as a fact in the program P. This is interesting since agents are reasoning about the world without depending on the actual situation going on, and thus leaving a more general program. Suppose for instance that only the first wise man has a white spot (i.e. $w_1 \wedge \neg w_2$) then he could deduce, just after seeing his partner, $\Box_1 w_1$. The model M discused above is no longer consistent with $P \cup \{\Box_1 w_1\}$ and, therefore, it should not be answer set.

This is exaclty the notion of nonmonotonic reasoning we are trying to model. It is, in some sense, safe for the second wise man to assume he has a white spot. Until he has evidence to believe the opposite, that is when the first wise man makes explicit his knowledge about the situation of the world, then he can provide an answer for sure. What we gain is the possibility to start making inference about the knowledge or beliefs of other agents without needing them to explicitly state such information.

Theorem proving tools of the Logics Workbench `LWB`[2], developed at the University of Bern in Switzerland, were used to check the proofs in this section.

[2] http://www.lwb.unibe.ch/

7 Conclusions

We propose how to do nonmonotonic reasoning using the modal logic S4. We also showed, generalizing a previous result in intuitionistic logic, how we can express the well known answer sets semantics using our approach. Observe that, in principle, it is possible to replace S4 with other stronger logics, up to S4.3, to get similar nonmonotonic systems. Interesting applications can also emerge if we allow the use of multimodal logic to model several interacting agents aimed with nonmonotonicity. Our results clearly state interesting links between ASP, modal and multi-modal systems, which might bring research of these areas together.

References

1. Michael Gelfond and Vladimir Lifschitz. Logic programs with classical negation. In David H. D. Warren and Péter Szeredi, editors, *Logic Programming, Proceedings of the Seventh International Conference*, pages 579–597, Jerusalem, Israel, June 1990. MIT Press.
2. Jan Jaspars. *Calculi for Constructive Communication, A study of the dynamics of partial states.* ITK and ILLC dissertation series, 1994.
3. Kurt Konolige. *A Deduction Model of Belief.* Morgan Kaufman Publishers, Inc., 1986.
4. Vladimir Lifschitz, David Pearce, and Agustin Valverde. Strongly equivalent logic programs. *ACM Transactions on Computational Logic*, 2:526–541, 2001.
5. Vladimir Lifschitz, L. R. Tang, and H. Turner. Nested expressions in logic programs. *Annals of Mathematics and Artificial Intelligence*, 25:369–389, 1999.
6. Mauricio Osorio and Juan Antonio Navarro. Modal logic $S5_2$ and FOUR (abstract). Contributed Talk at 2003 Annual Meeting of the Association for Symbolic Logic, Chicago, June 2003.
7. Mauricio Osorio, Juan Antonio Navarro, and José Arrazola. A logical approach for A-Prolog. In Ruy de Queiroz, Luiz Carlos Pereira, and Edward Hermann Haeusler, editors, *9th Workshop on Logic, Language, Information and Computation (WoLLIC)*, volume 67 of *Electronic Notes in Theoretical Computer Science*, pages 265–275, Rio de Janeiro, Brazil, 2002. Elsevier Science Publishers.
8. Mauricio Osorio, Juan Antonio Navarro, and José Arrazola. Applications of intuitionistic logic in answer set programming. *Theory and Practice of Logic Programming (TPLP)*, 4:325–354, 2004.
9. Mauricio Osorio, Juan Antonio Navarro, and José Arrazola. Safe beliefs for propositional theories. Accepted to appear at Annals of Pure and Applied Logic, 2004.
10. David Pearce. Answer set inference and nonmonotonic S4. In *Extensions of Logic Programming*, number 798 in LNAI. Springer, Berlin, 1994.
11. David Pearce. From here to there: Stable negation in logic programming. In D. M. Gabbay and H. Wansing, editors, *What Is Negation?*, pages 161–181. Kluwer Academic Publishers, Netherlands, 1999.
12. David Pearce. Stable inference as intuitionistic validity. *Logic Programming*, 38:79–91, 1999.
13. Grigori Schwarz and Miroslaw Truszczynski. Nonmonotonic reasoning is sometimes simpler! *Journal of Logic and Computation*, pages 295–308, 1996.