AUTOEPISTEMIC LOGICS AS A UNIFYING FRAMEWORK FOR THE SEMANTICS OF LOGIC PROGRAMS*

PIERO A. BONATTI

In this paper, it is shown that a three-valued autoepistemic logic provides an elegant unifying framework for some of the major semantics of normal and disjunctive logic programs and logic programs with classical negation, namely, the stable semantics, the well-founded semantics, supported models, Fitting's semantics, Kunen's semantics, the stationary semantics, and answer sets. For the first time, so many semantics are embedded into one logic. The framework extends previous results—by Gelfond, Lifschitz, Marek, Subrahmanian, and Truszczynski—on the relationships between logic programming and Moore's autoepistemic logic. The framework suggests several new semantics for negation-as-failure. In particular, we will introduce the epistemic semantics for disjunctive logic programs. In order to motivate the epistemic semantics, an interesting class of applications called ignorance tests will be formalized; it will be proved that ignorance tests can be defined by means of the epistemic semantics, but not by means of the old semantics for disjunctive programs. The autoepistemic framework provides a formal foundation for an environment that integrates different forms of negation. The role of classical negation and various forms of negation-by-failure in logic programming will be briefly discussed.

*This paper, which contains part of the author's Ph.D. Thesis, is an extended and revised version of [7].

Address correspondence to Piero A. Bonatti, Dipartimento di Informatica, Università di Torino, Corso Svizzera 185, I-10145 Torino, Italy. E-mail: bonatti@di.unito.it.
1. INTRODUCTION

This paper introduces a unifying framework for some of the major semantics of logic programs. It will be shown how the stable semantics, the well-founded semantics, supported models, Fitting's semantics, Kunen's semantics, answer sets, and the stationary semantics can all be captured by one three-valued autoepistemic logic. The framework extends previous results by Gelfond, Lifschitz, Marek, Subrahmanian, and Truszczynski on the relations between logic programming and autoepistemic logics [16, 24, 38].

The framework is motivated by the following considerations:

- A unifying framework for different semantics is expected to facilitate their comparison. It seems impossible to say something new in the area of normal programs; the mutual relations between their semantics have already been extensively investigated, from many points of view, in [24, 4, 36, 37] and many other papers. However, the unifying framework might help to compare the various semantics for disjunctive programs, which have not yet been investigated so deeply.¹

- The study of the relations between logic programs and autoepistemic theories shows how logic programming techniques can be used to implement subsets of autoepistemic logic. From another point of view, the same relations legitimize the use of negation-as-failure for knowledge representation and belief modeling.

- The framework suggests several new semantics for negation as failure that overcome some of the limitations of the old semantics.

- However, no single form of negation seems to be satisfactory for all applications. This fact is leading to a never ending proliferation of semantics. Apparently, it is always possible to find an example where the existing semantics do not have the desired behavior. This problem might be solved by more flexible logic programming languages that allow different forms of negation to be expressed at the same time. Among other benefits, such languages would yield a better trade-off between inferential power and computational complexity, by allowing the more powerful—hence more complex—semantics to be applied only to the subprograms that really need their power. The autoepistemic framework suggests how to obtain such a language: by allowing the belief operator to occur explicitly in rules, one can express many different forms of negation-as-failure (possibly new ones) and make use of many forms of negation in one program.

- Finally, the framework might help to answer two old questions: Is classical negation really needed in logic programs? What is the non-monotonic formalism underlying logic programming (or is there one)?

It is impossible to include in one paper a complete study of all these potential applications of the framework. Thus the paper focuses only on a few aspects. After an introduction to three-valued autoepistemic logics (Section 2), we will show how the semantics of normal programs can be captured by these logics (Section 4). Then we will focus our attention on disjunctive programs (Section 5). After

¹Another contribution to this area is due to Dix [10]. It is a classification based on abstract properties such as cumulativity.
showing how the stationary semantics is captured by three-valued autoepistemic logics, we will introduce a set of new semantics, called epistemic semantics, that are naturally suggested by the framework. In order to motivate the epistemic semantics, we will introduce the notion of ignorance tests. It will be shown that ignorance test can be easily defined through the epistemic semantics, while they cannot be satisfactorily defined by means of the existing forms of negation as failure, no matter what representation method is adopted. Section 6 outlines the relations between three-valued autoepistemic logics and answer sets. Finally, in Section 7, it will be shown that one of the three-valued autoepistemic logics is sufficient to capture all the semantics of the framework.

Several comments and short discussions about the other applications of the framework can be found across all sections, and especially in Section 8.

We assume that the reader is familiar with logic programming and autoepistemic logic. We refer to [21] for the basic notions about logic programming, and to [36, 37] for an interesting global view on the major semantics of negation-as-failure. We refer to [27] for the basic definitions and properties of autoepistemic logic. The proofs of the results on the strong and weak well-founded semantics of disjunctive programs assume notions (like active derivation, basis, global tree, level) that can be found in [35].

Some proofs and technical lemmas have been moved to the Appendix; such lemmas are numbered “A.n.”

2. THREE-VALUED AUTOEPISTEMIC LOGIC

It is well known that \{-Lp \rightarrow p\} has no stable expansions. The reason is that this set of premises is ambiguous:

- If \( p \) is not believed (i.e., \(-Lp\) is true), then \( p \) follows from the premises and hence \( p \) should be believed.
- Conversely, if \( p \) is believed (i.e., \( Lp \) is true), then \( p \) does not follow from the premises and hence \( p \) should not be believed.

In both cases, \( p \) should be believed and should not be believed at the same time. In other words, this set of premises provides reasons both for believing \( p \) and for not believing it.

Another famous example of ambiguous premises is \{-Lp \rightarrow q, \neg Lq \rightarrow p\}. These premises have two stable expansions: in one of them, \( p \) is believed; in the other one, \( p \) is not believed. In other words, these premises also provide reasons both for believing \( p \) and for not believing it.

A cautious agent should not make arbitrary decisions: if some ambiguous premises provide a reason to believe \( p \) and a reason for not believing \( p \), then \( p \) should be neither believed nor disbelieved. This is the main intuition underlying three-valued autoepistemic logics. The rest of this section is devoted to formalizing it. Some of the definitions and results contained in this section appeared in [5-7].

2.1. Definitions

The language of autoepistemic logic is a propositional modal language, \( L_L \), with standard connectives (\( \lor, \land, \rightarrow, \) and \( \neg \)) and one modal operator \( L \), to be read as “know” or “believe.”
The sentences where $L$ does not occur are called ordinary or objective. The set of ordinary sentences is denoted by $\mathcal{L}$. The sentences where propositional symbols occur only within the scope of $L$ are called subjective. The sentences of the form $L\psi$ (where $\psi$ is an arbitrarily complex $\mathcal{L}_L$ sentence) are called autoepistemic atoms. The following abbreviations will be used:

\[
\begin{align*}
\overline{X} &= \mathcal{L}_L \setminus X, \\
\text{Ord}(X) &= X \cap \mathcal{L}.
\end{align*}
\]

Autoepistemic logics model the beliefs that an ideally rational and introspective agent should hold, given a set $S$ of premises (i.e., axioms or basic beliefs). In Moore's logic, the beliefs of the agent are represented by means of a set of sentences. In the three-valued autoepistemic logic, a slightly more complex structure is needed in order to model doubtful agents.

**Definition 2.1.** A belief state $B$ is a pair $\langle B^+, B^- \rangle$, where $B^+$ and $B^-$ are disjoint sets of $\mathcal{L}_L$ sentences.

Intuitively, $B^+$ is the set of statements that are believed by the agent, while $B^-$ is the set of statements that the agent has no reason to believe. The remaining statements of the language are those about which the agent is doubtful, that is, such statements are involved in an ambiguous piece of knowledge.

We say that a belief state $B$ is complete iff $B^+ \cup B^- = \mathcal{L}_L$. Belief states are partially ordered by the natural extension of set inclusion, defined as follows:

\[
B_1 \subseteq B_2 \text{ iff } B_1^+ \subseteq B_2^+ \text{ and } B_1^- \subseteq B_2^-.
\]

Union and intersection are extended to belief states in a similar way. It is easy to verify that the set of belief states, extended with the above ordering, is a complete lower semilattice (see [1.2] for the definition). There is one minimal element $\bot = \langle \emptyset, \emptyset \rangle$, but many maximal elements, corresponding to complete belief states. Next we introduce the models of our language.

**Definition 2.2.** A propositional interpretation is a pair $(I, B)$, where $I$ is a classical (two-valued) interpretation of $\mathcal{L}$ and $B$ is a belief state.

A propositional interpretation $(I, B)$ is also called a $B$-interpretation. A $B$-interpretation is complete iff $B$ is complete.

Intuitively, in a propositional interpretation $(I, B)$, $I$ models what is true in the "outside world," while $B$ models the agent's beliefs. $I$ assigns a classical truth value to ordinary atoms, while $B$ defines a three-valued truth assignment to autoepistemic atoms. Nonatomic sentences are evaluated by extending strong Kleene's valuation to the connective $\leftarrow$, which is interpreted in a nonstandard way: "$\psi \leftarrow \sigma$" should be read as "If $\sigma$ is true, then also $\psi$ is true." With this definition, implication satisfies the implication theorem. As usual, implication coincides with classical implication when both $\psi$ and $\sigma$ are defined. This informal description is made precise by the following definition.

**Definition 2.3.** The valuation $V_{I, B}$ associated with a propositional interpretation $(I, B)$ is defined as follows:
If \( p \) is an ordinary atom,

\[
\forall_{I,B}(p) = \begin{cases} 
    \text{true}, & \text{if } I \models p, \\
    \text{false}, & \text{otherwise}; 
\end{cases}
\]

\[
\forall_{I,B}(L\phi) = \begin{cases} 
    \text{true}, & \text{if } \phi \in B^+, \\
    \text{false}, & \text{if } \phi \in B^-, \\
    \text{undefined}, & \text{otherwise}; 
\end{cases}
\]

\[
\forall_{I,B}(\neg \phi) = \begin{cases} 
    \text{true}, & \text{if } \forall_{I,B}(\phi) = \text{false}, \\
    \text{false}, & \text{if } \forall_{I,B}(\phi) = \text{true}, \\
    \text{undefined}, & \text{otherwise}; 
\end{cases}
\]

\[
\forall_{I,B}(\phi \land \psi) = \begin{cases} 
    \text{true}, & \text{if } \forall_{I,B}(\phi) = \forall_{I,B}(\psi) = \text{true}, \\
    \text{false}, & \text{if } \forall_{I,B}(\phi) = \text{false} \text{ or } \forall_{I,B}(\psi) = \text{false}, \\
    \text{undefined}, & \text{otherwise}; 
\end{cases}
\]

\[
\forall_{I,B}(\phi \lor \psi) = \begin{cases} 
    \text{true}, & \text{if } \forall_{I,B}(\phi) = \text{true} \text{ or } \forall_{I,B}(\psi) = \text{true}, \\
    \text{false}, & \text{if } \forall_{I,B}(\phi) = \forall_{I,B}(\psi) = \text{false}, \\
    \text{undefined}, & \text{otherwise}; 
\end{cases}
\]

\[
\forall_{I,B}(\phi \leftarrow \psi) = \begin{cases} 
    \text{true}, & \text{if } \forall_{I,B}(\phi) = \text{true} \text{ or } \forall_{I,B}(\psi) \neq \text{true}, \\
    \text{false}, & \text{otherwise}. 
\end{cases}
\]

Note that ordinary sentences are given a thoroughly classical semantics. An ordinary sentence \( \psi \) is a (classical) tautology if and only if \( \psi \) is true in all propositional interpretations. Note also that complete interpretations are classical: they map each sentence on either \text{true} or \text{false}.\(^2\)

Classical terminology and notation are extended in the obvious way: a propositional interpretation \((I,B)\) satisfies a sentence \( \psi \) (denoted \((I,B) \models \psi\)) iff \( \forall_{I,B}(\psi) = \text{true} \). In this case, \((I,B)\) is called a \( B \)-model (or simply a model) of \( \psi \), and we say that \( \psi \) is \( B \)-consistent. A set \( S \) of sentences \( \text{entails} \) a sentence \( \psi \) (denoted \( S \models \psi \)) iff \( \psi \) is true in every model of \( S \). A set \( S \) of sentences \( B \)-entails a sentence \( \psi \) (denoted \( S \models_B \psi \)) iff \( \psi \) is true in every \( B \)-model of \( S \).

A propositional interpretation can be a model of a belief state in several ways.

\textit{Definition 2.4.} \((I,B')\) is an \textit{autoepistemic interpretation} of a belief state \( B \) iff \( B' = B \). \((I,B')\) is an \textit{autoepistemic model} of \( B \) iff \( B' = B \) and \((I,B') \models B^+ \). If \( B \) has an autoepistemic model, we say that \( B \) is \textit{epistemically consistent}.\(^3\)

Now we are ready to specify how the agent should derive his/her belief state from the set of premises. The following discussion will introduce and motivate the three-valued counterparts of stable expansions. In the following, let \( S \) and \( B \) denote the premises and the belief state of the agent, respectively.

Essentially, the definition of stable expansions follows from the assumption that

\(^2\)The notion of propositional interpretation introduced by Moore [27] corresponds to our notion of complete propositional interpretation.

\(^3\)These are the obvious extensions of Moore's notions. An anonymous referee pointed out that the definition of autoepistemic model is asymmetric: there is no negative counterpart of condition \((I,B) \models B^+ \). The reason is that the sentences of \( B^- \) are not necessarily true according to the agent's knowledge, but they may well be true in the real world.
the agent is ideally rational and introspective. In Moore’s logic, rationality is formalized by requiring a sentence $\psi$ to be believed if and only if $\psi$ is a logical consequence of the premises and introspective knowledge. In the three-valued case, doubts need to be taken into account. Doubts are caused by ambiguous premises, that, for some sentence $\psi$, suggest both that $\psi$ should be believed and that $\psi$ should not be believed. Since the agent should make no arbitrary decision, his/her conclusions should not be influenced by doubts. This form of independence will be expressed as follows.

**Principle 1.** The agent should believe $\psi$ if $\psi$ is a consequence of the premises and introspective knowledge, no matter how the agent’s doubts can be removed.

**Principle 2.** The agent should have no reason to believe $\psi$ if $\psi$ is not a consequence of the premises and introspective knowledge, no matter how the agent’s doubts can be removed.

To formalize the above principles, one should specify how the agent’s doubts can possibly be removed. Equivalently, we will specify how $B$ can be extended to a new belief state $B' \supseteq B$. Such admissible extensions can be defined in several reasonable ways, which can be regarded as different assumptions on how the agent’s knowledge may evolve along time. According to the easiest definition, one may state that $B'$ is an admissible extension of $B$ whenever $B \subseteq B'$. Under this definition, a new piece of information that removes some doubts might contradict $S$.

**Example 2.1.** Let $S = \{ \neg p \leftarrow \neg Lp, \neg q \leftarrow \neg Lq, p \lor q \}$. $S$ has two consistent stable expansions that contain $p$ and $q$ (respectively) but not both. Therefore, $S$ is ambiguous; $p$ and $q$ should be neither believed not disbelieved and hence $p$ and $q$ should be neither in $B^+$ nor in $B^-$. Thus, among the possible extensions of $B$, we have $B' = \langle B^+, B^- \cup \{p, q\} \rangle$, which, roughly speaking, corresponds to a hypothetical situation where $\neg Lp, \neg Lq$ become true. Now, it is easy to see that $S$ is $B'$-inconsistent (i.e., for all $I : (I, B') \not\models S$. Consequently, if we regard $B'$ as an admissible (or possible) extension of the agent’s belief state, then we must implicitly accept that a new piece of knowledge (in this case, $\neg Lp, \neg Lq$) that removes some of the agent’s doubts, might simultaneously contradict the premises. In this case (as a consequence of the second principle), no sentence can be disbelieved, because every sentence follows from the premises and the new introspective knowledge. This is not unreasonable, for after detecting an inconsistency the agent should revise his/her beliefs (if possible) and hence, in general, one cannot say a priori which sentences will not be believed. However, the resulting logic is very weak: we have $B^- \neq \emptyset$, which means that no nonmonotonic deduction is made.\(^4\)

According to another definition, $B'$ can be an admissible extension of $B$ only if $S$ is $B'$-consistent, which reflects greater confidence in the premises, because it is implicitly assumed that doubts will be removed without contradicting $S$. Of course, this form of admissible extension solves the problem illustrated by the previous example (it guarantees that $B^- \neq \emptyset$). The formal definition is the following.

\(^4\)Several nonmonotonic logics have the same limitation. See the section on related work for more details.
Definition 2.5. The set of admissible extensions of $B$ (w.r.t. $S$) is

$$\mathcal{A}_S(B) = \{B' \mid B' \text{ is a belief state, } B' \supseteq B \text{ and } S \text{ is } B'\text{-consistent}\}.$$ 

Essentially, in this paper we will adopt this definition because it seems to provide the least inferential capabilities required to capture the semantics of disjunctive programs.\(^5\) Further notions of admissible extensions are studied in [6, 8]. Finally, we need the following constraint.

**Principle 3.** The agent's belief state should be epistemically consistent.

From a technical point of view, this constraint is needed to guarantee that the introspective knowledge of the agent exactly matches his/her belief state. However, the constraint can also be explained in terms of rationality and introspectiveness: An ideally introspective agent knows that the real world is an interpretation of the form $(I, B)$, where $B$ is exactly his/her belief state. If no such interpretation satisfied the agent's beliefs $(B^+)$, then the agent would have no reason to trust his/her beliefs. Therefore, we require that for some $I, (I, B) \models B^+$, which means that $B$ should be epistemically consistent.

The above principles are formalized by the following definition.

Definition 2.6. Let $S$ be a set of premises. A belief state $B$ is a generalized stable expansion (GSE) of $S$ iff:

1. $B^+ = \{\psi \mid \forall B' \in \mathcal{A}_S(B), S \models_{B'} \psi\}$.
2. $B^- = \{\psi \mid \forall B' \in \mathcal{A}_S(B), S \not\models_{B'} \psi\}$.
3. $B$ is epistemically consistent.

Every GSE is a possible belief state for our agent. GSE's enjoy a generalization of Stalnaker's stability conditions, namely, the agent’s beliefs are closed under logical consequence, and his/her introspective knowledge corresponds exactly to his/her belief state.

**Theorem 2.1 (Stability).** If $B$ is a generalized stable expansion, then $B$ is stable, that is:

(i) If $B^+ \models \psi$, then $\psi \in B^+$.
(ii) $L\psi \in B^+$ iff $\psi \in B^+$.
(iii) $\neg L\psi \in B^+$ iff $\psi \in B^-$.

GSE's can also be expressed in terms of a transformation over belief states.

Definition 2.7. Operator $\Theta_S = \langle \Theta_S(B)^+, \Theta_S(B)^- \rangle$ is defined by the following

\(^5\)In fact, the stationary schemata that will be employed to capture the stationary semantics may easily cause the problem illustrated by the previous example. On the contrary, when we restrict to the autoepistemic translations of normal programs, the two forms of admissible extension coincide (such translations are $B'$-consistent for all $B'$). Indeed, in [7], the first formulation was adopted.
equations:
\[
\Theta_S(B)^+ = \{ \psi \mid \forall B' \in \mathcal{A}_S(B), S \models_B \psi \},
\]
\[
\Theta_S(B)^- = \{ \psi \mid \forall B' \in \mathcal{A}_S(B), S \not\models_B \psi \}.
\]

From Definitions 2.6 and 2.7, it follows immediately that the GSE's of $S$ are the epistemically consistent fixed points of $\Theta_S$. The operator is monotonic.

**Lemma 2.1 (Monotonicity).** $B_1 \subseteq B_2$ implies $\Theta_S(B_1) \subseteq \Theta_S(B_2)$.

Therefore, by Tarski's theorem, $\Theta_S$ has a least fixed point, equal to $(\Theta_S \uparrow \alpha)$, for some ordinal $\alpha$. Obviously, if the least fixed point is epistemically consistent, then it is the least GSE of $S$. We will see that the least fixed point may not be epistemically consistent, because, premises may have more than one minimal GSE or no GSE at all.

### 2.2. Relations with Moore’s Logic

Stable expansions can be considered as a special case of GSE's.

**Theorem 2.2.** $T$ is a consistent stable expansion of $S$ iff $(T, \overline{T})$ is a complete GSE of $S$.

In other words, consistent stable expansions and complete GSE's are in one-to-one correspondence. On the contrary, incomplete GSE's may not correspond to any stable expansion (see next example). Moore's logic can be captured by means of the simple axiom schema

(\text{CA}) \quad L\psi \lor \neg L\psi

in the sense that the consistent stable expansions of $S$ and the GSE's of $S \cup \text{CA}$ are in one-to-one correspondence.

**Theorem 2.3.** If $T$ is a consistent stable expansion of $S$, then $(T, \overline{T})$ is a GSE of $S \cup \text{CA}$. Conversely, if $B$ is a GSE of $S \cup \text{CA}$, then $B^+$ is a consistent stable expansion of $S$ and $B^- = \overline{B^+}$.

Intuitively, CA force the agent to have no doubts, and hence eliminates all incomplete GSE's; the remaining GSE's correspond to stable expansions by Theorem 2.2.

Theorems 2.3, 2.2, and 2.1 show that the three-valued logic is a natural generalization of Moore's logic. Theorem 2.2 proves also that the three-valued logic is at least as "robust" as Moore's logic, in the sense that a set of premises has a consistent stable expansion only if it has a GSE. The converse is not true.

**Example 2.2.** Let $S = \{ p, q \leftarrow \neg Lq \}$. $S$ has no stable expansion, but it has a GSE $B$, such that $p \in B^+, q \not\in B^+$, and $q \not\in B^-$. In other words, the agent believes $p$ and is doubtful about $q$. 
An immediate consequence of Theorem 2.3 is that some premises have many minimal GSE’s and some have no GSE’s. In order to obtain some interesting results on the existence of GSE’s and least GSE’s we have to restrict our attention to a specific class of premises.

### 2.3 Implicative Premises

**Definition 2.8.** A sentence \( \varphi \) is implicative if \( \varphi \) is ordinary, or \( \varphi = (\varphi_1 \leftarrow \varphi_2) \), where \( \varphi_1 \) is an ordinary sentence, \( \varphi_2 \) is subjective, and \( \leftarrow \) does not occur in \( \varphi_2 \). A set of sentences is implicative if it contains only implicative sentences.

For example, \( p \lor \neg q \leftarrow \neg Lr \land Ls \) is implicative, while \( CA \) is not. Implicative premises are important per se, because they are expressive and enjoy a number of nice properties. In this paper, however, they are relevant mainly because they capture the common features of all the autoepistemic translations of logic programs.

For implicative premises we have a strong result:

**Theorem 2.4.** Let \( S \) be an implicative set of sentences and let \( B = lfp(\Theta_S) \). Then the following are equivalent:

(i) \( S \) has a GSE.

(ii) \( B^+ \) is consistent.

(iii) \( B \) is the least GSE of \( S \).

The above theorem can be rephrased by borrowing terminology from logic programming: implicative premises have a fixed-point semantics, which corresponds to the declarative semantics provided by their least GSE. However, unlike logic programs, implicative theories are expressive enough to be inconsistent, i.e., without GSE’s. Implicative theories can be understood in terms of classical logic. Define:

\[
\text{Heads}(S) = \{ \varphi \mid (\varphi \leftarrow \sigma) \in S \} \cup \text{Ord}(S), \\
\text{Active}(S, B) = \{ (\varphi \leftarrow \sigma) \mid (\varphi \leftarrow \sigma) \in S \text{ and } \models_B \sigma \} \cup \text{Ord}(S), \\
\text{Cons}(S, B) = \text{Heads} (\text{Active}(S, B)).
\]

For example, if \( S = \{ p, (q \leftarrow Lr \land \neg Ls), (a \leftarrow \neg Lb) \} \) and \( B = \{ \{ r \}, \{ s \} \} \), then

\[
\text{Heads}(S) = \{ p, q, a \}, \\
\text{Active}(S, B) = \{ p, q \leftarrow Lr \land \neg Ls \}, \\
\text{Cons}(S, B) = \{ p, q \}.
\]

Note that ordinary sentences are considered as the conclusions of implications whose body is always true. \( \text{Cons}(S, B) \) is a set of ordinary sentences; it captures the ordinary \( B \)-consequences of \( S \), as it is shown by point iv of the next lemma. Most of the other facts that are proved in the lemma are needed for technical reasons. The reader may focus attention on points (v) and (vi), which show how the ordinary part of \( \Theta_S \) can be derived by means of classical logic.

**Lemma 2.2 (Elementary properties of implicative theories).** If \( S \) is an implicative set of sentences and \( \varphi \) is an ordinary sentence, then:
(i) $B \subseteq B'$ implies $\text{Cons}(S, B) \subseteq \text{Cons}(S, B')$.
(ii) $(I, B) \models S$ iff $I \models \text{Cons}(S, B)$.
(iii) $B' \in \mathcal{A}_S(B)$ iff $B' \supseteq B$ and $\text{Cons}(S, B')$ is consistent.
(iv) $S \models B \varphi$ iff $\text{Cons}(S, B) \vdash \varphi$.
(v) $\varphi \in \Theta_S(B)^+$ iff $\text{Cons}(S, B) \vdash \varphi$.
(vi) $\varphi \in \Theta_S(B)^-$ iff for all $B' \supseteq B$, either $\text{Cons}(S, B')$ is inconsistent or $\text{Cons}(S, B') \not\vdash \varphi$.

Moreover, if $B$ is a fixed point of $\Theta_S$:

(vii) $\varphi \in B^+$ iff $\text{Cons}(S, B) \vdash \varphi$.
(viii) $\varphi \in B^-$ iff for all $B' \supseteq B$, either $\text{Cons}(S, B') \not\vdash \varphi$ or $\text{Cons}(S, B')$ is inconsistent.

Finally, we introduce a subclass of implicative premises which is characterized by a generalization of an important property of logic programs:

**Definition 2.9.** An implicative set of sentences $S$ is a **quasiprogram** iff $\text{Heads}(S)$ is (classically) consistent.

This property allows us to improve Theorem 2.4.

**Theorem 2.5.** If $S$ is a quasiprogram, then $\text{lfp}(\Theta_S)$ is the least GSE of $S$.

We will see that many of the autoepistemic translations of logic programs are quasiprograms.

### 3. LOGIC PROGRAMS

A **normal program** is a set of rules having the form

$$A \leftarrow A_1, \ldots, A_n, \sim A_{n+1}, \ldots, \sim A_m,$$

where $A, A_1, \ldots, A_m$ are atoms. A **disjunctive program** is a set of rules having the form

$$A_1 \lor \cdots \lor A_k \leftarrow A_{k+1}, \ldots, A_n, \sim A_{n+1}, \ldots, \sim A_m.$$

In this paper we restrict our attention to (possibly infinite) propositional programs. This assumption is not really restrictive. In fact, all the semantics dealt with in this paper give each program $P$ and its ground instantiation $P'$ the same meaning, and $P'$ can be translated into a propositional theory by uniformly translating ground atoms into propositional symbols.

Therefore, assume a fixed propositional language $\mathcal{L}$ and denote with $\mathcal{H}$ the set of atoms (or propositional symbols) of $\mathcal{L}$. $\mathcal{H}$ is usually called Herbrand base and is only required to be a countable set. The following abbreviations will be needed:

$-X = \mathcal{H}/X,$

$\text{Atom}(X) = \mathcal{H} \cap X.$

Atom ($\cdot$) will be extended to belief states in the obvious way. $\text{Lit}(X)$ will denote the set of literals of $X$. Also $\text{Lit}(\cdot)$ will be extended to belief states in the
obvious way. The notion of Herbrand base has to be generalized in order to fit the expressiveness of disjunctive programs.

Definition 3.1. The conjunctive Herbrand base \([2]\), denoted \(\text{CHB}\), is the set of all nonredundant conjunctions of ordinary atoms.

The disjunctive Herbrand base \([2]\), denoted \(\text{DHB}\), is the set of all nonredundant disjunctions of ordinary atoms.

We will need a uniform way to represent different semantics for logic programs in order to facilitate their comparison.

Definition 3.2. A semantics is a partial function from the set of normal programs into belief states.

Therefore, if \(\text{SEM}\) is a semantics and \(\text{SEM}(P)\) is defined, then \(\text{SEM}(P)\) is a pair of sets of sentences, that will be denoted

\[(\text{SEM}(P)^+, \text{SEM}(P)^-)\].

In the semantics proposed so far, \(\text{SEM}(P)^+\) is the set of true sentences, while \(\text{SEM}(P)^-\) is the set of false sentences. This naturally leads to the following definition.

Definition 3.3. We say that a semantics \(\text{SEM}\) is consistent if, when \(\text{SEM}(P)\) is defined, there is a model of \(\text{SEM}(P)^+\) that falsifies all the sentences of \(\text{SEM}(P)^-\).

4. THE SEMANTICS OF NORMAL PROGRAMS

In this section, let the metavariable \(P\) range over normal programs.

4.1. Autoepistemic Semantics

The relations between logic programming and autoepistemic logics were explored for the first time by Gelfond \([14]\). He defined an autoepistemic semantics for logic programs by the following method:

- Each program \(P\) is translated into a set of autoepistemic sentences \(P_{b1}\).
- If \(P_{b1}\) has a unique stable expansion \(T\), then the atomic part of \(T\) provides the canonical model of \(P\).

More specifically, Gelfond adopted the following translation.

Definition 4.1 (Autoepistemic translation \(P_{b1}\) \([14]\)). \(P_{b1}\) is obtained by translating each rule \(A \leftarrow A_1, \ldots, A_n, \sim A_{n+1}, \ldots, \sim A_m\) of \(P\) into

\[A \leftarrow A_1 \land \cdots \land A_n \land \neg A_{n+1} \land \cdots \land \neg A_m.\]

In this section, Gelfond’s method will be extended to the three-valued autoepistemic logic and to other translations, in order to capture some of the major semantics of normal programs in one scheme. For each semantics, we will recall its
definition and show its relations with the three-valued autoepistemic logic. Finally, a new semantics, induced by the unifying schema, will be briefly discussed.

4.2. Stable Semantics

In [16], Gelfond’s autoepistemic semantics has been entirely rephrased in terms of logic programming concepts. This led to the notion of stable model.

Definition 4.2 (Gelfond–Lifschitz transformation, $\Pi_P$, stable models [16]). Let $P$ be a normal program and $I$ be a Herbrand interpretation. The Gelfond–Lifschitz transformation of $P$ with respect to $I$ is the normal program $P^I$, obtained from $P$ by:

1. Eliminating every clause whose body contains a negative literal $\neg A$, where $A \in I$,
2. Deleting all negative literals from the body of the remaining clauses.

Denote by $lm(P^I)$ the least model of $P^I$ and define

$$\Pi_P(I) = lm(P^I).$$

An Herbrand interpretation $I$ is a stable model of a normal program $P$ iff $I$ is a fixed point of $\Pi_P$.

Two different semantics based on stable models have been proposed [16, 37]:

Definition 4.3.

$$ST^!(P) = \begin{cases} \{I, \neg I\}, & \text{if } I \text{ is the only stable model of } P, \\ \text{undefined,} & \text{otherwise}; \end{cases}$$

$$ST(P) = \begin{cases} \bigcap\{\{I, \neg I\} | I \text{ is a stable model of } P\}, & \text{if } P \text{ has a stable model,} \\ \text{undefined,} & \text{otherwise.} \end{cases}$$

The next lemma shows the similarity between the operator $\Pi_P$ used in the definition of stable models and $\Theta_{P_{b1}}$.

Lemma 4.1. For all complete belief states $B$,

$$\Pi_P(Atom(B^+)) = Atom(\Theta_{P_{b1}}(B^+)).$$

The stable models of $P$ are in one-to-one correspondence with the complete GSE’s of $P_{b1}$.

Theorem 4.1. $I$ is a stable model of $P$ if and only if $I = Atom(B^+)$, where $B$ is a complete GSE of $P_{b1}$.

Proof. It is well known that $I$ is a stable model of $P$ iff $I = Atom(T)$, where $T$ is a stable expansion of $P_{b1}$ (see [16]). By Theorem 2.2, $T$ is a stable expansion of $P_{b1}$.
iff \((T, \overline{T})\) is a complete GSE of \(P_{b1}\). The Theorem follows by letting \(B = (T, \overline{T})\).  

As an immediate corollary, the two stable semantics can be expressed in terms of the three-valued autoepistemic logic.

**Corollary 4.1.**

\[
ST(P) = \begin{cases} 
\text{Atom}(B), & \text{if } B \text{ is the only complete GSE of } P_{b1}; \\
\text{undefined}, & \text{otherwise}; 
\end{cases}
\]

\[
ST(P) = \begin{cases} 
\bigcap \{\text{Atom}(B) \mid B \text{ is a complete GSE of } P_{b1}\}, & \text{if } P_{b1} \text{ has a complete GSE}; \\
\text{undefined}, & \text{otherwise}. 
\end{cases}
\]

### 4.3. Well-Founded Semantics

In the last subsection we have proved that the complete (and hence maximal) GSE's of \(P_{b1}\) correspond to the stable models of \(P\). This raises two natural questions: Is there a least GSE of \(P_{b1}\)? Which is the corresponding semantics for logic programs? The answer to the first question is yes.

**Proposition 4.1.** \(lfp(\Theta_{P_{b1}})\) is the least GSE of \(P_{b1}\).

**Proof.** \(P_{b1}\) is equivalent to a quasiprogram, obtained by replacing each rule

\[A \leftarrow A_1 \land \cdots \land A_n \land \neg L A_{n+1} \land \cdots \land \neg L A_m\]

with the equivalent implicative rule

\[(A \leftarrow A_1 \land \cdots \land A_n) \leftarrow \neg L A_{n+1} \land \cdots \land \neg L A_m.\]

It follows easily, by Theorem 2.5, that \(lfp(\Theta_{P_{b1}})\) is the least GSE of \(P_{b1}\).  

The second question will be answered by showing that the least GSE of \(P_{b1}\) corresponds to the well-founded semantics of \(P\).

The well-founded semantics has been introduced in [39] and has been reformulated many times ([28, 40, 37], etc.). None of these formulations will be illustrated here. We prefer to characterize the well-founded semantics as the least fixed point of an operator \(\Psi_P\) derived from \(\Pi_P\), because this formulation will make the proof of the correspondence theorem much simpler. The proof that the definition presented here is equivalent to the original one can be found in Appendix B.

**Definition 4.4.** For all Herbrand belief states \(B\) define

\[\Psi_P(B) = (\Pi_P(-B^-), -\Pi_P(B^+)).\]

The well-founded semantics of \(P\) (denoted by \(WF(P)\)) is the least fixed point of \(\Psi_P\).
The interested reader may easily verify that the fixed points of $\Psi_P$ correspond to the three-valued stable models of $P$ \cite{30}. $\Psi_P$ is the exact counterpart of $\Theta_{P_{51}}$.

**Lemma 4.2.** For all belief states $B$, $\Psi_P(\text{Atom}(B)) = \text{Atom}(\Theta_{P_{51}}(B))$.

Lemma 4.2 provides a representation of $\Theta_{P_{51}}$ in terms of $\Pi_P$, i.e., in terms of logic programming notions. As an immediate consequence of Lemma 4.2 and of the monotonicity of $\Theta_{P_{51}}$ (Lemma 2.1), we have that $\Psi_P$ is monotonic, too.

**Lemma 4.3.** For all Herbrand belief states $B_1, B_2, B_1 \subseteq B_2$ implies $\Psi_P(B_1) \subseteq \Psi_P(B_2)$.

Thus, by Tarski's theorem, $\text{lfp}(\Psi_P)$ always exists and equals $\Psi_P \uparrow \alpha$, for some ordinal $\alpha$. Now we are ready to prove that the least GSE of $P_{51}$ corresponds to the well-founded semantics.

**Theorem 4.2.** $WF(P) = \text{Atom}(B)$, where $B$ is the least GSE of $P_{51}$.

**Proof.** As a consequence of Tarski's theorem, there exists an ordinal $\alpha$ such that $WF(P) = \Psi_P \uparrow \alpha$ and $B = \Theta_{P_{51}} \uparrow \alpha$. Moreover, for all $\beta$, we have $\Psi_P \uparrow \beta = \text{Atom}(\Theta_{P_{51}} \uparrow \beta)$ (it follows from Lemma 4.2, through an easy induction on $\beta$). Conclude that $WF(P) = \Psi_P \uparrow \alpha = \text{Atom}(\Theta_{P_{51}} \uparrow \alpha) = \text{Atom}(B)$. $\square$

### 4.4. Supported Models

Apt, Blair, and Walker \cite{1} introduced the following notion.

**Definition 4.5.** A Herbrand interpretation $I$ is a supported model of $P$ iff for all $a \in I$, there exists a rule $a \leftarrow \psi$ in $P$ such that $I \models \psi$.

The correspondence between supported models and autoepistemic logic was investigated by Marek and Subrahmanian \cite{24}. It was observed in \cite{25} that their result can be restated in terms of the following translation.

**Definition 4.6 (Autoepistemic translation $P_{52}$).** $P_{52}$ is obtained by translating each rule $A \leftarrow A_1, \ldots, A_n, \neg A_{n+1}, \ldots, \neg A_m$ of $P$ into

$$A \leftarrow L A_1 \land \cdots \land L A_n \land \neg L A_{n+1} \land \cdots \land \neg L A_m.$$  

**Theorem 4.3** (\cite{24, 25}). $I$ is a supported model of $P$ iff $I = \text{Atom}(T)$, where $T$ is a stable expansion of $P_{52}$.

From this theorem and Theorem 2.2 we immediately get the relations between supported models and three-valued autoepistemic logics:

**Theorem 4.4.** $I$ is a supported model of $P$ iff $I = \text{Atom}(B^+)$, where $B$ is a complete GSE of $P_{52}$.

Note the analogy between this result and Theorem 4.1. The interested reader
may verify that $\Theta_{P_2}$ is similar to the immediate-consequences operator $T_P$, that is, for all complete belief states $B$,

$$T_P(\text{Atom}(B^+)) = \text{Atom}(\Theta_{P_2}(B))^+.$$

In the rest of the paper, we will need the following semantics, based on supported models:

$$\text{SUPP}(P) = \begin{cases} 
\bigcap\{\{I,-I\} \mid I \text{ is a supported model of } P\}, & \text{if } P \text{ has a supported model,} \\
\text{undefined,} & \text{otherwise.}
\end{cases}$$

4.5. Fitting's Semantics

Note that $P_2$ is a quasiprogram and, therefore, by Theorem 2.5, it always has one minimal GSE. We are going to investigate the semantics induced by such GSE; it will turn out to be very similar to the Kripke–Kleene semantics introduced by Fitting [13].

Fitting’s semantics is captured by the least fixed point of a suitable operator, denoted by $\Phi_P$. Fitting’s definition employs annotated sentences. For the sake of simplicity, the essentially equivalent definition proposed by Kunen is adopted here:

**Definition 4.7 ([19])**. For all Herbrand belief states $B$ define

$$\Phi_P(B)^+ = \{ A \mid \text{for some clause} \left( A \leftarrow A_1, \ldots, A_n, \sim A_{n+1}, \ldots, \sim A_m \right) \text{ in } P \}
\quad A_1, \ldots, A_n \in B^+ \text{ and } A_{n+1}, \ldots, A_m \in B^- \}.$$

$$\Phi_P(B)^- = \{ A \mid \text{for all clauses} \left( A \leftarrow A_1, \ldots, A_n, \sim A_{n+1}, \ldots, \sim A_m \right) \text{ in } P \}
\quad \text{there is an atom } A_i \text{ such that either}
\quad \text{there is an atom } A_i \text{ such that either}
\quad i \leq n \text{ and } A_i \in B^- \text{ or } i > n \text{ and } A_i \in B^+ \}.$$

Fitting's semantics is $\text{FIT}(P) = \text{lfp}(\Phi_P)$.

The following result proves that the least GSE of $P_2$ is closely related to Fitting’s semantics.

**Theorem 4.5**. Let $P$ be any program and let $P'$ be the program obtained from $P$ by eliminating all the clauses that contain a pair of complementary literals in the body. Finally, let $B$ be the least GSE of $P_2$. Then

$$\text{Atom}(B) = \text{FIT}(P').$$

Intuitively, the semantics induced by $P_2$ ignores all the clauses of the form

$$A \leftarrow \cdots A_i \cdots \sim A_i \cdots$$

because their body can never be satisfied.
Example 4.1. Let $P = \{(p \leftarrow q, \neg q), (q \leftarrow \neg q)\}$. In $FIT(P)$, $p$ is undefined because both $q$ and $\neg q$ are undefined. On the contrary, $p$ is false in the semantics induced by $P_{b_2}$, because the first rule of $P$ is ignored. According to this semantics, $P$ is equivalent to $\{q \leftarrow \neg q\}$.

If no rule contains a pair of complementary literals in its body, then Fitting's semantics corresponds exactly to the least GSE of $P_{b_2}$. The semantics induced by $P_{b_2}$ does not correspond to any known semantics; however, the possibility of making inferences, like deducing $\neg p$ from the program $P$ of Example 1, is considered as a desirable property.

Fitting's semantics is not exactly captured by $P_{b_2}$ because, for every atom $A$, the translations of $A$ and $\neg A$ are mutually inconsistent. This can be avoided in several ways by exploiting the intensional nature of the belief operator $L$. Here we adopt the following translation.

Definition 4.8 (Autoepistemic translation $P_{b_3}$). $P_{b_3}$ is obtained by translating each rule $A \leftarrow A_1, \ldots, A_n, \neg A_{n+1}, \ldots, \neg A_m$ of $P$ into

$$A \leftarrow LLA_1 \land \cdots \land LLA_n \land L\neg LA_{n+1} \land \cdots \land L\neg LA_m.$$  

This translation maps $A$ and $\neg A$, respectively, into $LLA$ and $L\neg LA$, which are mutually consistent. As a result, we get the desired correspondence between $\Phi_P$ and $\Theta_{P_{b_3}}$.

Lemma 4.4. For all $B$, $\Phi_P(\text{Atom}(B)) = \text{Atom}(\Theta_{P_{b_3}}^2(B))$.

Note that $\Phi_P$ corresponds to two applications of $\Theta_{P_{b_3}}$; this is due to the nesting of the modal operator in $P_{b_3}$. From Lemma 4.4, it follows that Fitting's semantics is the atomic part of the least GSE of $P_{b_3}$.

Theorem 4.6. $FIT(P) = \text{Atom}(B)$, where $B$ is the least GSE of $\Theta_{P_{b_3}}$.

Proof (Similar to the proof of Theorem 4.2). As a consequence of Tarski's theorem, there exists an ordinal $\alpha$ such that $FIT(P) = \Phi_P \uparrow \alpha$ and $B = \Theta_{P_{b_3}} \uparrow \alpha = \Theta_{P_{b_3}}^2 \uparrow \alpha$. Moreover, for all $\beta$, we have $\Phi_P \uparrow \beta = \text{Atom}(\Theta_{P_{b_3}}^2 \uparrow \beta)$ (it follows from Lemma 4.4, through an easy induction on $\beta$). Conclude that $FIT(P) = \Phi_P \uparrow \alpha = \text{Atom}(\Theta_{P_{b_3}}^2 \uparrow \alpha) = \text{Atom}(B)$. \[ \square \]

We are left to talk about the semantics induced by the complete GSE's of $P_{b_3}$. They correspond to supported models. In fact, as the interested reader may easily verify, the stable expansions of $P_{b_3}$ coincide with the stable expansions of $P_{b_2}$, since their rules are equivalent in every stable set. As a consequence, also the complete GSE's of $P_{b_3}$ and $P_{b_2}$ coincide. By Theorem 4.4, they all correspond to supported models.

4.6. Kunen's Semantics

Kunen's semantics can be obtained by truncating the inductive construction of Fitting's semantics after $\omega$ steps [19].
Definition 4.9. \( K\text{UN}(P) = \Phi_P \uparrow \omega \).

It follows easily from Lemma 4.4 that \( \Phi_P \uparrow \omega = \text{Atom}(\Theta_{P_{\text{b}}} \uparrow \omega) \). Thus we derive the following theorem.

Theorem 4.7. For all normal programs \( P \),
\[
K\text{UN}(P) = \text{Atom}(\Theta_{P_{\text{b}}} \uparrow \omega).
\]

Therefore, Kunen’s semantics can be expressed in terms of the autoepistemic operator \( \Theta_{P_{\text{b}}} \). When \( \Phi_P \) is continuous (e.g., when \( P \) has a finite Herbrand base) \( K\text{UN}(P) = F\text{IT}(P) \). Thus we get the following theorem.

Theorem 4.8. If \( \Phi_P \) is continuous, then
\[
K\text{UN}(P) = \text{Atom}(B),
\]
where \( B \) is the least GSE of \( P_{\text{b}} \).

5. THE SEMANTICS OF DISJUNCTIVE PROGRAMS

In this section, \( P \) will range over disjunctive programs. Among the first extensions of the well-founded semantics to disjunctive programs we find the strong, weak, and optimal well-founded semantics for disjunctive programs introduced by Ross [35]. Other major approaches are the stationary semantics [29], the generalized well-founded semantics for disjunctive programs (GDWFS) [2], and its extension \( W\text{F}^3 \) [3].

The three well-founded semantics by Ross and the stationary semantics are consistent, in the sense of Definition 3.3. On the contrary, GDWFS and \( W\text{F}^3 \) have recently been shown to map some programs onto inconsistent belief states [11]. Consistency is an important property for the aforementioned semantics, because the sentences of \( SEM(P)^- \) are assumed to be false on the basis of nonmonotonic inferences. We say that these semantics are based on a notion of negation-as-assumed-falsity.

In Section 5.2 some interesting nonconsistent semantics (based on a different notion of negation) will be introduced. However, first we will focus our attention on the stationary semantics and show how it can be captured by the three-valued autoepistemic logic.

5.1. The Stationary Semantics

The stationary semantics is essentially based on the extended closed world assumption (ECWA). Define
\[
ECWA(S) = \{ \psi \mid \psi \text{ is true in all the minimal models of } S \}.
\]

In order to define the stationary semantics, the language has to be extended with a set of new atoms, \( \mathcal{H}' \), such that;

- \( \mathcal{H} \cap \mathcal{H}' = \emptyset \).
• For each atom $p$ in $\mathcal{H}$ there exists a distinct atom $Bp$ in $\mathcal{H}'$.

For all disjunctive programs $P$ (built from the atoms of $\mathcal{H}$) define a new program $P_{pB}$ by translating each negative literal $\neg a$ into $\neg Ba$. Then define the following sequence:

$$S_0 = \emptyset,$$
$$S_{\alpha+1} = \{Bp_1 \lor \cdots \lor Bp_n \mid p_1 \lor \cdots \lor p_n \in ECWA(P_{pB} \cup S_\alpha)\}$$
$$\cup \{-Bp_1 \lor \cdots \lor -Bp_n \mid \neg p_1 \lor \cdots \lor \neg p_n \in ECWA(P_{pB} \cup S_\alpha)\},$$
$$S_\lambda = \bigcup_{\alpha < \lambda} S_\alpha,$$

where $\lambda$ ranges over limit ordinals. The sequence is monotonic and reaches a limit at some ordinal $\delta$, that is, for all $\alpha > \delta, S_\alpha = S_\delta$. $S_\delta$ is called the stationary state of $P$. The stationary state is the basis of the stationary semantics.

**Definition 5.1 ([29]).**

$$STN(P)^+ = \{p_1 \lor \cdots \lor p_n \mid Bp_1 \lor \cdots \lor Bp_n \in S_\delta\},$$
$$STN(P)^- = \{p_1 \land \cdots \land p_n \mid \neg Bp_1 \land \cdots \land \neg Bp_n \in S_\delta\},$$

where $S_\delta$ is the stationary state of $P$.

In order to express the stationary semantics through three-valued autoepistemic logic, it suffices to add to $P_{pB}$ the following axiom schemata, that we call stationary schemata:

$$\neg p_1 \lor \cdots \lor \neg p_n \leftarrow \neg L(p_1 \land \cdots \land p_n),$$
$$Bp_1 \lor \cdots \lor Bp_n \leftarrow L(p_1 \lor \cdots \lor p_n),$$
$$\neg Bp_1 \lor \cdots \lor \neg Bp_n \leftarrow L(\neg p_1 \lor \cdots \lor \neg p_n).$$

Intuitively, the first schema corresponds to the ECWA for conjunctions (i.e., each conjunction that is not derivable should be assumed to be false), while the other schemata transform the atoms of $\mathcal{H}$ into the corresponding atoms of $\mathcal{H}'$, as it happens when $S_{\alpha+1}$ is generated from $S_\alpha$.

In the following, the set of instances of the stationary schemata will be denoted by $SA$.

The reader will not be surprised to see that the least GSE of $P_{pB} \cup SA$ corresponds to the stationary semantics. In the following, let $DC(B)$ be an abbreviation for

$$\langle B^+ \land DHB, B^- \land CHB \rangle.$$
Theorem 5.1. Let $P$ be a disjunctive program and let $B = \text{lfp} \left( \Theta_{P_u \cup \text{SA}} \right)$.

(i) $B$ is the least GSE of $P_u \cup \text{SA}$.
(ii) $\text{STN}(P) = \text{DC}(B)$.

5.2. Epistemic Semantics and Pure Negation-as-Failure

In this section we introduce some interesting semantics which are inconsistent in the sense of Definition 3. The basic idea is that the autoepistemic translations of normal programs can immediately be extended to disjunctive programs. For example, if $P = \{a \lor b \leftarrow p, \neg q\}$, then

\[
P_{b1} = \{a \lor b \leftarrow p \land \neg Lq\},
\]
\[
P_{b2} = \{a \lor b \leftarrow Lp \land \neg Lq\},
\]
\[
P_{b3} = \{a \lor b \leftarrow L\neg Lp \land L\neg Lq\}.
\]

Thus each of the semantics for normal programs captured by the autoepistemic framework has a natural disjunctive counterpart, based on the same autoepistemic translation and on the same class of GSE’s.

Definition 5.2. Epistemic semantics. For all disjunctive programs $P$ define

\[
\text{STN}_{\text{DIS}}(P) = \begin{cases} 
\text{DC}(B), & \text{if } B \text{ is the only complete GSE of } P_{b1}, \\
\text{undefined}, & \text{otherwise}; 
\end{cases}
\]

\[
\text{STN}_{\text{DIS}}(P) = \begin{cases} 
\bigcap \{\text{DC}(B) \mid B \text{ is a complete GSE of } P_{b1}\}, & \text{if } P_{b1} \text{ has a complete GSE,} \\
\text{undefined}, & \text{otherwise}; 
\end{cases}
\]

\[
\text{WF}_{\text{DIS}}(P) = \text{DC}(B_{P_{b1}}), \text{ where } B_{P_{b1}} \text{ is the least GSE of } P_{b1};
\]
\[
\text{FIT}_{\text{DIS}}(P) = \text{DC}(B_{P_{b3}}), \text{ where } B_{P_{b3}} \text{ is the least GSE of } P_{b3};
\]
\[
\text{KUN}_{\text{DIS}}(P) = \text{DC}(\Theta_{P_{b3}} \uparrow \omega).
\]

Note the analogy with Corollary 4.1 and with Theorems 4.2, 4.6, and 4.7. Note also that $P_{b1}$ and $P_{b3}$ are quasiprograms and, hence, by Theorem 2.5, $\text{lfp}(\Theta_{P_{b1}})$ and $\text{lfp}(\Theta_{P_{b3}})$ are the least GSE’s of $P_{b1}$ and $P_{b3}$, respectively. Therefore, $\text{WF}_{\text{DIS}}$ and $\text{FIT}_{\text{DIS}}$ are always defined and enjoy a fixed-point semantics given by the ordinal sequences $(\Theta_{P_{b1}} \uparrow 0), (\Theta_{P_{b1}} \uparrow 1), \ldots$ and $(\Theta_{P_{b3}} \uparrow 0), (\Theta_{P_{b3}} \uparrow 1), \ldots$.

The following example shows that none of the epistemic semantics is consistent.

Example 5.1. Let $P = \{p \lor q\}$. Note that $P_{b1} = P_{b3} = P$. $P$ and its translations have one GSE, denoted by $B$, which corresponds to the unique stable expansion of $P$, that is, $B = \langle E(P), E(P) \rangle$.\footnote{E(·) maps each set of ordinary sentences into its unique stable expansion which is defined in [23].} Obviously, $p \lor q \in B^+$, while $p, q \in B^-$: therefore, $B$ is inconsistent.
Inconsistency is not a problem because—according to the epistemic semantics—\( \neg A \) should be read as “\( A \) is not derivable” or “\( A \) is not believed” rather than “\( A \) is false.” We call this form of negation pure negation-as-failure. It is easy to see that the epistemic semantics are different from the semantics based on negation-as-assumed-falsity.

**Example 5.2.** Let \( P \) and \( B \) be defined as in Example 5.1 and let \( SEM \) range over the various well-founded semantics by Ross and over the stationary semantics, GDWFS, and WF\(^3\). \( SEM(P) \) is consistent and \( p,q \notin SEM(P)^- \). It follows immediately that \( SEM(P) \neq B \), i.e, \( SEM \) is different from all the epistemic semantics.

For the program illustrated in the above examples, the epistemic semantics are strictly stronger than the other semantics. However, in general, the semantics belonging to the two classes are not comparable.

**Example 5.3.** Let \( P = \{(a \leftarrow \neg p, \neg q), (p \lor q)\} \). According to the epistemic semantics, \( a \) can be derived from \( P \), because neither \( p \) nor \( q \) can be derived and hence \( \neg p \) and \( \neg q \) both follow from \( P \).

On the contrary, according to the other semantics (excepting the weak well-founded semantics), \( a \) is false (i.e., \( a \notin SEM(P)^- \)) because the body of the corresponding rule is inconsistent with \( p \lor q \). According to the weak well-founded semantics, instead, \( a \) is undefined.

\( FITDIS \) and \( KUNDIS \) seem to be of little interest because their inferences are extremely weak: rules can never derive new facts from any disjunction.

**Example 5.4.** Let \( P = \{(a \leftarrow p), (a \leftarrow q), (p \lor q)\} \). One would expect \( a \) to be derivable from this program. However, \( a \) is not believed in the least GSE of \( P_{53} \). More specifically, we have \( P_{53} = \{(a \leftarrow LLp), (a \leftarrow LLq), (p \lor q)\} \). Neither \( p \) nor \( q \) are derivable and hence \( LLp \) and \( LLq \) are false. It follows that \( a \in FITDIS(P)^- \). The same is true of \( KUNDIS \), because, in this case, \( KUNDIS(P) = FITDIS(P) \).

For this reason, \( FITDIS \) and \( KUNDIS \) will not be further discussed. We will focus our attention on the other epistemic semantics. The following examples illustrate these semantics and show that the main characteristics of \( WF, ST! \), and \( ST \), as well as their mutual relations, are preserved by their disjunctive counterparts.

**Example 5.5.** Let \( P \) be the following normal program:

\[
\begin{align*}
  a & \leftarrow \neg b, \\
  b & \leftarrow \neg a, \\
  r & \leftarrow a, \neg r, \\
  p & \leftarrow b.
\end{align*}
\]

Then we have

\[
ST!(P) = ST(P)
\]
\[
= \langle \{b, p\}, \{a, r\} \rangle,
\]
\[
WF(P) = \langle \emptyset, \emptyset \rangle.
\]

Note the peculiar conclusions of \(STI\) and \(ST\), that cannot be reached through \(WF\). Now consider the disjunctive program \(P'\), obtained from \(P\) by replacing \(p\) with \(p_1 \vee p_2\), that is,
\[
a \leftarrow \sim b,
\]
\[
b \leftarrow \sim a,
\]
\[
r \leftarrow a, \sim r.
\]
\[
p_1 \vee p_2 \leftarrow b.
\]

\(STI_{DIS}(P'), ST_{DIS}(P'\rangle\), and \(WF_{DIS}(P'\rangle\) are similar to \(STI(P), ST(P),\) and \(WF(P),\) respectively. We have
\[
STI_{DIS}(P) = ST_{DIS}(P)
\]
\[
= \langle \|b, p_1 \vee p_2\|, \|a, r, p_1, p_2\| \rangle.
\]
\[
WF(P) = \langle \emptyset, \|p_1, p_2\| \rangle.
\]

where \(\|d_1, \ldots, d_n\|\) is the set of nonredundant disjunctions of atoms that contain some \(d_i\) and \(\|c_1, \ldots, c_n\|\) is the set of nonredundant conjunctions of atoms that contain some \(c_i\).

\(p_1\) and \(p_2\) have no counterpart in \(P\). They are disbeliefed because there is no way to derive them—at most, in some generalized stable expansions, it is possible to believe \(p_1 \vee p_2\).

Example 5.6. Let \(P\) be the following normal program:
\[
a \leftarrow \sim b,
\]
\[
b \leftarrow \sim a.
\]
\[
p \leftarrow a.
\]
\[
r \leftarrow p.
\]
\[
r \leftarrow b.
\]

Then we have
\[
STI(P) = undefined.
\]
\[
ST(P) = \langle \{r\}, \emptyset \rangle.
\]
\[
WF(P) = \langle \emptyset, \emptyset \rangle.
\]

Note that \(r\) can be concluded by means of \(ST\), but not by means of the other semantics. This is a typical nonconstructive inference of \(ST\), called floating con-
clusion. Now consider the disjunctive program $P'$, obtained from $P$ by replacing $p$ with $p_1 \lor p_2$, that is,

\[
\begin{align*}
a & \leftarrow \neg b, \\
b & \leftarrow \neg a, \\
p_1 \lor p_2 & \leftarrow a, \\
r & \leftarrow p_1, \\
r & \leftarrow p_2, \\
r & \leftarrow b.
\end{align*}
\]

$ST_{DIS}(P')$ and $WF_{DIS}(P')$ are similar to $ST(P)$ and $WF(P)$ respectively. In fact,

\[
\begin{align*}
ST_{DIS}(P') & = \text{undefined}, \\
WF_{DIS}(P') & = (\emptyset, |p_1, p_2|).
\end{align*}
\]

Also $ST_{DIS}(P')$ and $ST(P)$ are similar, although their comparison is slightly less obvious. We have

\[
ST_{DIS}(P') = \langle r, (a \lor b), (p_1 \lor p_2 \lor b)\rangle, |p_1, p_2|).
\]

Note that all the disjunctions in $ST_{DIS}(P')^+$ are either explicitly or implicitly true in $ST(P)$.

\begin{itemize}
  \item $r$ is both in $ST_{DIS}(P')^+$ and $ST(P)^+$.
  \item $a \lor b$ is true in all the stable models of $P$.
  \item $p_1 \lor p_2 \lor b$ corresponds to $p \lor b$, which is true in all the stable models of $P$.
\end{itemize}

Strange as they may seem, the semantics based on pure negation-as-failure prove to be indispensable for a large class of applications, where certain inferences have to be triggered by partial lack of knowledge. The following are two examples of such applications.

**Example 5.7.** A doctor knows that his patient is affected either by disease $d_1$ or by disease $d_2$. If the doctor knows which disease is affecting the patient, then she/he can apply a specific therapy. If the doctor does not know which disease is affecting the patient, then she/he should infer that more clinical tests are needed, because prescribing the wrong treatment may be dangerous for the patient.

This example can be formalized by the following program, where $t_1, t_2$ represent the therapies for $d_1$ and $d_2$, respectively, and $t_3$ represents the need for clinical tests:

\[P = \{(t_1 \leftarrow d_1), (t_2 \leftarrow d_2), (t_3 \leftarrow \neg d_1, \neg d_2)\}.
\]

If an epistemic semantics is adopted, then $P$ is a correct formalization of the example, in the following sense:

\begin{itemize}
  \item $t_1 \in SEM(P \cup \{d_1\})^+$ and $t_2, t_3 \in SEM(P \cup \{d_1\})^-$. \\
  \item $t_2 \in SEM(P \cup \{d_2\})^+$ and $t_1, t_3 \in SEM(P \cup \{d_2\})^-$. \\
  \item $t_3 \in SEM(P \cup \{d_1 \lor d_2\})^+$ and $t_1, t_2 \in SEM(P \cup \{d_1 \lor d_2\})^-$. \\
\end{itemize}
$P$ is correct also in a more general sense: $t_1, t_2$, and, most of all, $t_3$ have their intended meaning not only in $P \cup \{d_1\}, P \cup \{d_2\}$, and $P \cup \{d_1 \lor d_2\}$, but also in every program $P \cup EDB$, where $EDB$ is a set of positive clauses that do not contain $t_1, t_2, t_3$.

**Example 5.8.** Consider a census database $DB$ such that, for all persons $x$, $\text{male}(x) \lor \text{female}(x)$ is a logical consequence of $DB$. At the time of the creation of $DB$, the data of some persons may be partially unavailable; for example, the sex of some individual $x$ may not be known. In order to collect the missing information, it may be helpful to make a list of all the persons whose data are not complete. For this purpose one should define a relation $r$ such that $r(x)$ is inferred from $DB$ if and only if neither $\text{male}(x)$ nor $\text{female}(x)$ can be derived from $DB$.

Note the similarity between this example and the problem of finding null values of a certain kind, namely, values which are only partially specified by a finite range of possible values, which can be represented by a disjunction like $\psi(r_1) \lor \cdots \lor \psi(r_n)$ (see Reiter [34]).

Under the epistemic semantics, the relation $r$ can be defined by the simple program

$$P = \{ r(x) \leftarrow \neg \text{male}(x), \neg \text{female}(x) \}.$$  

If some form of negation-as-asserted-falsity is adopted, then the formalizations illustrated in the previous examples are not correct: $t_3$ turns out to be false in $P \cup \{d_1 \lor d_2\}$; similarly, $r(x)$ is false when the database contains $\text{male}(x) \lor \text{female}(x)$. Indeed, negation-as-asserted-falsity makes it extremely hard to define relations like $t_3$ and $r$, as the reader may easily verify. In order to give a formal explanation of this fact, we will define a class of relations—called ignorance tests—by abstracting the common features of $t_3$ and $r$.

### 5.3. Ignorance Tests

First we recall the standard notions of extensional and intensional databases. Let $\mathcal{H}$ be partitioned in two sets: the set of extensional atoms and the set of intensional atoms. An *extensional database* is a set of facts of the form

$$A_1 \lor \cdots \lor A_n,$$

where the $A_i$'s are extensional atoms. An *intensional database* is a set of rules of the form

$$A_1 \lor \cdots \lor A_k \leftarrow A_{k+1}, \ldots, A_n, \neg A_{n+1}, \ldots, A_m,$$

where $A_1 \cdots A_k$ are intensional atoms.

Now we are ready to define ignorance tests. For the sake of simplicity, we give the definition only for total semantics.

---

8 Of course, both $r$ and $t_3$ (from the previous example) can be easily defined with any language that contains a belief operator $L$ (e.g., Gelfond's epistemic specifications [15] and Przymusinski's epistemic logic [33]) by replacing $\neg$ with $\neg L$. 

Definition 5.3. Let \( p, q \) be extensional atoms. An *ignorance test for \( p, q \)* is a triple \( (P, r, SEM) \), where \( P \) is a disjunctive program, \( r \) is an intensional atom, and \( SEM \) is a total semantics, and such that for all extensional databases \( EDB \),

\[
r \in SEM(P \cup EDB)^+ \iff EDB \not\models p \text{ and } EDB \not\models q.
\]

Ignorance tests generalize the essential features of Examples 5.7 and 5.8. In the former, \( (P, t_3, SEM) \) is an ignorance test for \( d_1 \) and \( d_2 \), provided that \( SEM \) is an epistemic semantics. Under the same condition, in Example 5.8, \( (P, r(t), SEM) \) is an ignorance test for \( \text{male}(t) \) and \( \text{female}(t) \), for all ground terms \( t \).

Ignorance tests are difficult to define by means of negation-as-falsity, because the former are inherently nonmonotonic, as it is shown by the following lemma.

Lemma 5.1. Let \( p, q \) be extensional atoms, let \( SEM \) be a total semantics, and let \( P \) be a disjunctive program. If there exists an extensional database \( EDB \) such that:

1. \( EDB \not\models p \text{ and } EDB \not\models q \), and
2. either \( SEM(P \cup EDB) \subseteq SEM(P \cup EDB \cup \{p\}) \) or \( SEM(P \cup EDB) \subseteq SEM(P \cup EDB \cup \{q\}) \),

then, for all intensional atoms \( r \), \( (P, r, SEM) \) is not an ignorance test for \( p, q \).

Proof. Straightforward from the definition of ignorance tests. \( \square \)

On the contrary, the known forms of negation-as-assumed-falsity exhibit a monotonic behavior in a large number of cases. Consider the weak well-founded semantics, for example:

Lemma 5.2. Let \( SEM \) be the weak well-founded semantics for disjunctive programs and let \( p, q \) be distinct atoms. Then, for all \( P \),

\[
SEM(P \cup \{p \lor q\}) \subseteq SEM(P \cup \{p \lor q\} \cup \{p\}).
\]

The above two lemmas imply that, by means of the weak well-founded semantics, it is impossible to define any ignorance test. This is formalized by the following theorem.

Theorem 5.2. Let \( SEM \) be the weak well-founded semantics. For all \( P, r, \) and \( SEM \), and for all distinct extensional atoms \( p \) and \( q \), \( (P, r, SEM) \) is not an ignorance test for \( p \) and \( q \).

Proof. Straightforward from Lemmas 5.2 and 5.1. \( \square \)

The strong well-founded semantics suffers from similar problems. It is always possible to extend any \( EDB \) in a way that causes the well-founded semantics to behave monotonically. This fact is formalized by the next lemma.

Lemma 5.3. Let \( SEM \) be the strong well-founded semantics, let \( P \) be a disjunctive program, and let \( p, q, c \) be distinct atoms, where \( c \) does not occur in the heads of
the rules of $P$. Finally, let $P' = P \cup \{p \lor q, p \lor c\}$. Then some of the following conditions hold:

- $SEM(P') \subseteq SEM(P' \cup \{p\})$.
- $SEM(P') = SEM(P' \cup \{q\})$.

From the above result and Lemma 5.1 we immediately derive the following theorem.

**Theorem 5.3.** Let $SEM$ be the strong well-founded semantics. $\langle P, r, SEM \rangle$ can be an ignorance test for $p, q$ only if each atom of $H$ occurs in the head of some rule of $P$.

**Proof.** Suppose not, i.e., there exists an ignorance test $\langle P, r, SEM \rangle$ and an atom $c$ that does not occur in the head of any rule of $P$. Let $EDB = \{(p \lor q), (p \lor c)\}$. By Lemma 5.3, at least one of the following inclusions holds:

$$SEM(P \cup EDB) \subsetneq SEM(P \cup EDB \cup \{p\}),$$
$$SEM(P \cup EDB) \subseteq SEM(P \cup EDB \cup \{q\})$$

and hence, by Lemma 5.1, $\langle P, r, SEM \rangle$ cannot be an ignorance test, which contradicts the assumptions. \[\square\]

Therefore, in general, it is impossible to write a program incrementally, because any axiomatization of ignorance test must take into account all the extensional atoms, and hence the set of extensional predicates should be fixed in advance. For the same reason, whenever a database update introduces a new atom, it might be necessary to rewrite all ignorance tests. We conclude that, in general, ignorance tests cannot satisfactorily be defined by means of the strong well-founded semantics.

The same difficulties are encountered with the stationary semantics. The next definitions and Theorem 5.4 characterize a large number of cases where $STN$ is monotonic.

**Definition 5.4.** A Przymusinski state (P-state for short) is a set of clauses each of which has one of the following forms:

$$BP_1 \lor \cdots \lor BP_n,$$
$$\neg BP_1 \lor \cdots \lor \neg BP_n,$$

where $BP_i \in H'(i = 1, \ldots, n)$.

**Definition 5.5.** Let $p, q$ be atoms and let $T$ be a set of sentences. Let $MM(X)$ denote the set of minimal models of $X$. $p$ and $q$ are independent in $T$ iff, for all $P$-states $S$:

$$MM(T \cup S) = MM(T \cup \{p\} \cup S) \cup MM(T \cup \{q\} \cup S).$$

\[^9\text{Przymusinski states are called simply "states" in [29]. Here the name is changed in order to avoid confusion with the other uses of the word "state."}\]
Theorem 5.4 (Monotonicity of STN.). If \( p, q \) are independent in \( P_{P,B} \), then
\[
STN(P) \subseteq STN(P \cup \{p\}).
\]

Given a program \( P \) and a pair of atoms \( p \) and \( q \), it is very easy to extend \( P \) in such a way that \( p \) and \( q \) become independent:

Lemma 5.4. Let \( p, q, c, c' \) be distinct atoms of \( \mathcal{H} \), with the exception of \( c \) and \( c' \), which may be equal. Let \( P \) be a disjunctive program, whose rules contain neither \( c \) nor \( c' \) in their heads. Then \( p, q \) are independent in
\[
P' = [P \cup \{(p \lor q), (p \lor c), (q \lor c')\}]_{P,B}.
\]

As a consequence, we can extend Theorem 5.3 to the stationary semantics.

Theorem 5.5. \((P, r, STN)\) can be an ignorance test for \( p, q \) only if each atom of \( \mathcal{H} \) occurs in the head of some rule of \( P \).

Proof. Suppose not, i.e., there exists an ignorance test \((P, r, STN)\) for \( p, q \), and two atoms \( c \) and \( c' \) (possibly \( c = c' \)) that do not occur in the head of any rule of \( P \). Let \( EDB = \{(p \lor q), (p \lor c), (p \lor c')\} \). By Lemma 5.4, \( p, q \) are independent in \([P \cup EDB]_{P,B}\). By Lemma 5.4 we get
\[
STN(P \cup EDB) \subseteq STN(P \cup EDB \cup \{p\})
\]
and hence, by Lemma 5.1, \((P, r, STN)\) cannot be an ignorance test for \( p, q \), which contradicts the assumptions. \(\square\)

The discussion following Theorem 5.3 can be extended to \( STN \). We conclude that ignorance tests cannot satisfactorily be defined with the stationary semantics.

It remains to be seen whether GDWFS, WF\(^3\), and the many other semantics which have been introduced so far (e.g., \([30, 11, 10]\)) suffer from the same limitations. We leave this as an open problem. It is not yet known whether Theorem 5.3 can be extended to these semantics. It seems reasonable to conjecture that the same difficulties encountered with the strong well-founded and with the stationary semantics arise also with the other forms of negation-as-assumed-falsity.

Another interesting open problem is the following: Is it at all possible to define an ignorance test with the strong well-founded semantics and \( STN \)? Theorems 5.3 and 5.5 are all we have, and they tell nothing about the existence of such a test.

6. LOGIC PROGRAMS WITH CLASSICAL NEGATION

Gelfond and Lifschitz \([17]\) proposed a language based on literals of the form \( a \) or \( \neg a \) (where \( a \) is an atom) and on rules of the form
\[
l_0 \leftarrow l_1, \ldots, l_n, \neg l_{n+1}, \ldots, \neg l_m,
\]
where \( l_0, \ldots, l_m \) are literals. In their framework, stable models are replaced by answer sets, which are sets of literals. The definition of answer sets can be obtained
from the definition of stable models by replacing all the occurrences of "atom" with "literal." There is one further important condition: Every answer set which contains two complementary literals should contain all literals. If a program has one answer set \( S \), then \( S \) is the semantics of the program, which is undefined otherwise.

**Definition 6.1.** \( \text{ANS}^!(P) = (S, \text{Lit}(S)) \) if \( S \) is the only answer set of \( P \); otherwise, \( \text{ANS}^!(P) \) is undefined.

By analogy with the stable semantics, we can also define an alternative semantics:

**Definition 6.2.** \( \text{ANS}(P) = \bigcap \{(S, \text{Lit}(S)) | S \text{ is an answer set of } P \} \) if \( P \) has an answer set; otherwise, \( \text{ANS}(P) \) is undefined.

Answer sets can be captured in several ways:

- One might translate each negative literal \( \neg p \) into a new atom \( p^* \) and transform the resulting normal program \( P^* \) into \( P_{\beta 1} \cup \text{CA} \cup \text{Contr} \), where \( \text{Contr} \) is the following axiom schema:
  \[
  p \leftarrow q \land q^*.
  \]
- One might extend the semantics of three-valued autoepistemic logics by allowing three-valued outside worlds and use simply \( P_{\beta 1} \).
- One may exploit the relationships between answer sets and N-expansions [38], and between the latter and stable expansions [18].

In this paper we adopt the last approach, which is based on the following embedding.

**Definition 6.3 (Autoepistemic translation \( P_{\beta 4} \) [38]).** \( P_{\beta 4} \) is obtained by translating each rule \( A \leftarrow A_1, \ldots, A_n, \neg A_{n+1}, \ldots, \neg A_m \) of \( P \) into

\[
A \leftarrow LA_1 \land \cdots \land LA_n \land L \neg LA_{n+1} \land \cdots \land L \neg LA_m.
\]

The correspondence between the semantics of \( P \) and \( P_{\beta 4} \) is stated by the following theorem.

**Theorem 6.1.** Let \( P \) be a finite propositional logic program with classical negation and let \( M \) be a set of literals. \( M \) is a consistent answer set for \( P \) iff \( M = \text{Lit}(B^+) \), where \( B \) is a complete GSE of \( P_{\beta 4} \cup \{\text{LG}(P_{\beta 4})\} \) and \( \text{LG}(P_{\beta 4}) \) is the subjective sentence introduced by Gottlob [18].

**Proof.** By a result of [38], we have that \( M \) is an answer set for \( P \) iff \( M = \text{Lit}(T) \), where \( T \) is an N-expansion of \( P_{\beta 4} \). Moreover, by a result of [18], we have that \( T \) is an N-expansion of \( P_{\beta 4} \) iff \( T \) is a stable expansion of \( P_{\beta 4} \cup \{\text{LG}(P_{\beta 4})\} \). It follows, by Theorem 2.2, that \( M \) is an answer set for \( P \) iff \( M = \text{Lit}(B^+) \), where \( B \) is a complete GSE of \( P_{\beta 4} \cup \{\text{LG}(P_{\beta 4})\} \). \( \square \)

A similar result can be obtained by means of translation \( P_{\beta 3} \):

**Corollary 6.1.** Let \( P \) be a finite propositional logic program with classical negation...
and let \( M \) be a set of literals. \( M \) is an answer set for \( P \) iff \( M = \text{Lit}(B^+) \), where \( B \) is a complete GSE of \( P_{b3} \cup \{LG(P_{b4})\} \).

**Proof (Sketch).** First note that \( P_{b3} \cup \{LG(P_{b4})\} \) and \( P_{b4} \cup \{LG(P_{b4})\} \) have the same stable expansions, because \( P_{b3} \) and \( P_{b4} \) are equivalent in all stable sets. Then \( P_{b3} \cup \{LG(P_{b4})\} \) and \( P_{b4} \cup \{LG(P_{b4})\} \) have the same complete GSE’s. The corollary follows immediately from this fact and Theorem 6.1. \( \square \)

7. **THE AUTOEPISTEMIC LOGIC UNDERLYING LOGIC PROGRAMMING**

The autoepistemic framework introduced in the previous sections is actually composed of two distinct autoepistemic logics, due to the fact that stable models, supported models, and answer sets correspond to complete (and hence maximal) GSE’s, while the other semantics correspond to minimal GSE’s. Since the two criteria for selecting the GSE’s are expressed at the metalevel, they give rise to two different logics. However, stable and supported models and answer sets are captured also by the minimal GSE’s of suitable translations.

**Theorem 7.1.**

(i) \( I \) is a stable model of \( P \) if and only if \( I = \text{Atom}(B^+) \), where \( B \) is a minimal GSE of \( P_{b1} \cup CA \).

(ii) \( I \) is a supported model of \( P \) if and only if \( I = \text{Atom}(B^+) \), where \( B \) is a minimal GSE of \( P_{b2} \cup CA \).

(iii) \( M \) is an answer set for \( P \) if and only if \( M = \text{Lit}(B^+) \), where \( B \) is a minimal GSE of \( P_{b3} \cup CA \cup \{LG(P_{b4})\} \).

**Proof.** (i) \( I \) is a stable model of \( P \) iff \( I = \text{Atom}(B^+) \), where \( B \) is a complete stable expansion of \( P_{b1} \) (Theorem 4.1). Moreover, by Lemma A.4, \( B \) is a complete stable expansion of \( P_{b1} \) iff \( B \) is a minimal GSE of \( P_{b1} \cup CA \). Point (i) follows immediately.

(ii) and (iii) Similar (apply Theorem 4.4 and Corollary 6.1 instead of Theorem 4.1). \( \square \)

This fact implies that all the semantics captured by the unifying framework can be expressed by means of one logic, based on minimal GSE’s only.

**Definition 7.2.**

\[
3AEL(S) = \begin{cases} 
\bigcap \{B \mid B \text{ is a GSE of } P\}, & \text{if } P \text{ has a GSE}, \\
\text{undefined}, & \text{otherwise}.
\end{cases}
\]

The relations between logic programs and autoepistemic logics can be rephrased as follows:

\[
ST!(P) = \begin{cases} 
\text{Atom}(3AEL(P_{b1} \cup CA)), & \text{if } 3AEL(P_{b1} \cup CA \text{ is complete}}, \\
\text{undefined}, & \text{otherwise},
\end{cases}
\]

\[
ST(P) = \text{Atom}(3AEL(P_{b1} \cup CA)).
\]
\[
WF(P) = \text{Atom}(3AEL(P_{1})), \\
SUPP(P) = \text{Atom}(3AEL(P_{2} \cup CA)) \\
= \text{Atom}(3AEL(P_{3} \cup CA)), \\
FIT(P) = \text{Atom}(3AEL(P_{3})), \\
ANS!(P) = \begin{cases} 
\text{Lit}(3AEL(P_{3} \cup CA \cup \{LG(P_{4})\})), & \text{if it is complete,} \\
\text{undefined}, & \text{otherwise,}
\end{cases} \\
ANS(P) = \text{Lit}(3AEL(P_{3} \cup CA \cup \{LG(P_{4})\})), \\
STN(P) = DC(3AEL(P_{3} \cup SA)).
\]

The epistemic semantics can be rephrased as follows:

\[
ST_{DIS}(P) = \begin{cases} 
DC(3AEL(P_{1} \cup CA)), & \text{if } 3AEL(P_{1} \cup CA) \text{ is complete,} \\
\text{undefined}, & \text{otherwise,}
\end{cases}
\]

Embedding all semantics into one logic makes it easier to integrate them. The advantages of integrating different semantics are discussed in the next section.

8. DISCUSSION: WHAT NEGATION?

The inability of the old semantics to define ignorance tests does not imply that pure negation-as-failure is always better than negation-as-assumed-falsity. Unfortunately, I have not found a counterpart of ignorance tests, i.e., a class of interesting applications that allows us to prove formally that pure negation-as-failure alone is not enough—although one may reasonably conjecture that there exist some applications that can be naturally formulated with negation-as-assumed-falsity but not with pure negation-as-failure. Nevertheless, the fact that all the semantics proposed so far are based on negation-as-assumed-falsity shows that this kind of negation is felt to be natural by many researchers—and, indeed, the examples that motivated the old semantics are convincing. This observation suggests that a flexible logic programming language should embody both forms of negation.

The three-valued autoepistemic logic provides an appealing formal foundation for such a language, because it captures in a natural way both the stationary semantics and the new epistemic semantics.

Another reason for integrating different negations is that a flexible language allows us to improve the trade-off between inferential power and computational complexity. For example, it is well known that:

\[
FIT(P) \subseteq WF(P) \subseteq ST(P)
\]

(provided that \(ST(P)\) is defined, of course). Thus, the stable semantics can make more inferences than the other semantics, but the cost of this power is high. In fact, \(FIT(P)\) and \(WF(P)\) can be computed efficiently (in linear and quadratic time, respectively) while \(ST(P)\) is co-NP complete. By assigning different semantics to
distinct program modules, it may be possible to improve efficiency and to achieve a satisfactory inferential power at the same time.

Note that there is no reason to commit oneself to a fixed set of nonmonotonic negations. As it was suggested in [20], classical negation and the modal operator can be taken as the basic constructs. They can be used to mimic existing negations and to create new ones. This led to the proposal of autoepistemic logic programming language [20, 9].

A more extended analysis of the advantages of integrating different semantics and the development of integration methods lie beyond the scope of this paper. More details can be found in [9].

Another language, equivalent to the autoepistemic language can be obtained by replacing \( L \) with a form of pure negation-as-failure, denoted by \( \sim \), whose intended meaning is \( \neg L \). For example, an implication like \( a \leftarrow Lb \land \neg Lc \) should be represented as \( a \leftarrow \neg \neg b \land \sim c \). Each of the semantics for normal programs can be obtained by a combination of pure negation-as-failure and classical negation, where the latter needs to be applied only to literals of the form \( \sim \phi \). The stationary semantics, instead, needs classical negation to be applied also to ordinary atoms. The stationary axioms become

\[
\neg p_1 \lor \cdots \lor \neg p_n \leftarrow \sim (p_1 \land \cdots \land p_n),
\]

\[
Bp_1 \lor \cdots \lor Bp_n \leftarrow \sim (p_1 \lor \cdots \lor p_n),
\]

\[
\neg Bp_1 \lor \cdots \lor \neg Bp_n \leftarrow \sim (\neg p_1 \lor \cdots \lor \neg p_n).
\]

The fact that the stationary semantics can be expressed by a combination of classical negation and pure negation-as-failure (that seem to be more elementary than negation-as-assumed-falsity) raises an important question: Should a logic programming language be based on pure negation-as-failure and negation-as-assumed falsity, or should the latter rather be replaced by classical negation? This problem is not only of theoretical interest, since nonmonotonic negation is in general more complex than classical negation (e.g., \( ST \) is co-NP-complete, in the finite propositional cases [25], and it is not computable in the general case). This observation suggests that classical negation may be preferable to nonmonotonic negation, whenever nonmonotonic inferences are not really needed.

The above discussion provides new motivations for the study of logic programming languages with classical negation. An objection is that classical negation, as proposed in [17], is not essential, since it might be simulated by translating negative literals into new atoms. Roughly speaking, this is possible because classical negation—as defined in [17]—cannot be used to make nonconstructive inferences. This kind of reasoning can only be accomplished with nonmonotonic negation, through floating conclusions. The above considerations about complexity suggest a different role for classical negation, namely, it might be profitable to use it for nonconstructive reasoning, leaving nonmonotonic negations for the cases where nonmonotonic assumptions are really needed. This topic lies beyond the scope of the paper and will not be further discussed.
9. RELATED WORK

9.1. Other Formalizations of Doubts

A three-valued autoepistemic logic was introduced by Przymusinski [31] with a very different purpose, namely, showing that the well-founded semantics corresponds to suitable three-valued versions of the major nonmonotonic formalisms. No unifying frameworks for the semantics of logic programs were proposed in [31]. The relations between Przymusinski's three-valued autoepistemic logic and other semantics have not been investigated.

From a technical point of view, the main differences between Przymusinski's logic and my three-valued autoepistemic logic lie in the semantics of ordinary sentences: in the former logic, such sentences are given a three-valued semantics, while in the latter their semantics is classical. Consequently, ordinary tautologies hold in my logic, while they do not hold in Przymusinski's logic. In particular, the tautological schema \( p \lor \neg p \) (where \( p \) ranges over propositional symbols) plays in his logic the same role as \( CA \), namely, it makes the three-valued logic collapse to Moore's logic.

Przymusinski introduced also a family of autoepistemic formalisms, the logics of closed beliefs [32], where the agent may have doubts, although the underlying models are two-valued. He proved that the stable and the well-founded semantics correspond to two different logics of closed beliefs. However, the logics of closed beliefs have not been proposed as a unifying framework, and the relations with other semantics have not been investigated. A major difference between the logics of closed beliefs and the other autoepistemic formalisms (including our logics) is due to the fact that the agent's disbeliefs are derived through some closed 'world assumption, which gives the logics of closed beliefs some of the features of circumscriptive.

9.2. Strong and Biased Autoepistemic Logics

In his thesis [20], Kuo proposes autoepistemic logics as a unifying framework for the semantics of logic programs and as a tool for deriving new semantics. His framework is based on strong autoepistemic logic and biased autoepistemic logic. The latter can be regarded as a family of logics, generated by different bias orderings. The different semantics of logic programs are captured by different translations and different logics. Kuo's framework covers some semantics which are not considered in this paper, like the upper well-founded semantics and programs with exceptions. On the other hand, Kuo's framework does not cover Kunen's semantics nor the semantics of disjunctive programs.

9.3. Stable Classes

Baral and Subrahmanian [4] generalized default and autoepistemic logics by replacing the notion of fixpoint with the notion of stable class, which collapses to the notion of cycle in the finite propositional case. In this way, they obtained well-founded versions of default and autoepistemic logics. Recently, Yuan [41] introduced a similar framework for autoepistemic logic that combines aspects of stable classes and N-expansions. The major semantics captured by the two frameworks are the stable, well-founded, and stable class semantics for normal programs. Both frameworks are based on two-valued models.
The main advantage of three-valued autoepistemic logic over two-valued approaches (including the above two approaches, Kuo's framework and autoepistemic logics of closed beliefs) is that different logics can be captured within the logic, through simple axioms like CA. Thus three-valued autoepistemic logic induces flexible logic programming/knowledge representation languages that make it possible to choose and/or combine different reasoning modes without resorting to extralogical features. On the contrary, in the aforementioned two-valued approaches, different logics can be captured only at the metalevel, by imposing constraints on stable expansions or stable classes. The reason is that two-valued approaches have little or no ability to represent introspective knowledge about doubts.

It is worth noting that Yuan's logic and the well-founded formalisms by Baral and Subrahmanian can be regarded as two-valued, strongly grounded counterparts of the three-valued autoepistemic logic based on the simplest form of admissible extension:

\[ A_3(B) = \{ B' \mid B' \text{ is a belief state and } B' \supseteq B \} \]

Accordingly, Yuan's logic and well-founded autoepistemic logic have the limitation illustrated in Example 2.1: when classical negation occurs in the premises, the logic easily collapses to monotonic modal system N. For instance, \{ \neg p \leftarrow \neg Lp, \neg q \leftarrow \neg Lq, p \lor q \} has one generalized stable belief (i.e., Yuan's counterpart of GSE's) that does not allow us to deduce \neg L\phi for any \phi, including the propositional symbols that do not occur in the premises.

9.4. Formalisms Based on Two Modalities

Both the language of epistemic specifications introduced by Gelfond [15] and the language of Przymusinski's epistemic logic [33] are essentially based on two modalities:

- Epistemic specifications contain negation as failure (\neg), which can be regarded as a disbelief operator, and a modal operator K which can be regarded as a metabelief operator.
- The language of epistemic logic contains two classes of special atoms (of the form $\mathcal{L}p$ and $\mathcal{D}p$, respectively) which capture autoepistemic beliefs and truth in minimal models.

Thanks to double modalities, the two formalisms are very powerful and expressive, but, in general, they seem no easier to understand than 3AEL: epistemic specifications are given their meaning by two nested fixpoint constructions, corresponding to "$\neg$" and "$K$", respectively. Theories in epistemic logic are given a meaning through a fixpoint equation where autoepistemic introspection and minimal entailment are simultaneously performed. It is not difficult to see that different modalities may interact in subtle ways, and as usual, in order to prove interesting properties of theories, such as consistency and iterative characterizations, one has to restrict attention to theories that are closely related to logic programs and allow only limited forms of interaction between the two modalities.

The author believes that a serious comparison of the above formalisms with three-valued autoepistemic logic can be achieved only through formal techniques, such as formal characterizations of their expressive power or techniques analogous to ignorance tests. However, we may sketch a preliminary comparison.
A possible limitation of epistemic specifications has been pointed out by Przy- 
musinski [33]: apparently, they cannot capture the well-founded semantics.

Epistemic logic and three-valued autoepistemic logic capture almost the same 
semantics of normal and disjunctive logic programs (it is not yet clear whether the 
disjunctive partial stable semantics can be captured by \(3AEL\) and whether Fitting’s 
semantics and Kunen’s semantics can be captured by epistemic logic). Nonetheless, 
we can draw a subtle distinction between the two approaches: Przymusinski’s main 
goal is providing a simple, direct way of expressing both autoepistemic reasoning 
and model minimization, while one of the goals of the author is analyzing the 
relationships between the two forms of reasoning by defining model minimization 
in terms of the belief operator (through the stationary schemata). Epistemic logic 
says little about such relations, for model minimization is expressed through ad hoc 
constructs.

10. CONCLUSIONS

We have shown how one three-valued autoepistemic logic, called \(3AEL\), can capture 
most of the major semantics of logic programs. Although many unifying frameworks 
have been proposed since our three-valued autoepistemic logic was introduced, its 
positive features are not subsumed in those of any other formalism.

The main relations between three-valued autoepistemic logics and logic programs 
are summarized in Table 1. The autoepistemic framework has suggested many new 
semantics; some of them, like \(FIT_{DIS}\) and \(KUN_{DIS}\), are apparently too weak to 
be useful; some others, like the semantics induced by the least GSE of \(P_2\), and

**TABLE 1. Summary of the Main Relations.**

<table>
<thead>
<tr>
<th>Translation</th>
<th>Complete GSE's</th>
<th>Min. GSE's</th>
<th>(3AEL)</th>
<th>(\text{ifp}(\Theta_T))</th>
<th>(\Theta_T \uparrow \omega)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(P_{b1})</td>
<td>Stable models</td>
<td>(WF)</td>
<td>(WF)</td>
<td>(WF)</td>
<td>(WF)</td>
</tr>
<tr>
<td>(P_{b2})</td>
<td>Supported models</td>
<td>(WF)</td>
<td>(WF)</td>
<td>(WF)</td>
<td>(WF)</td>
</tr>
<tr>
<td>(P_{b3})</td>
<td>Supported models</td>
<td>(FIT)</td>
<td>(FIT)</td>
<td>(FIT)</td>
<td>(KUN)</td>
</tr>
<tr>
<td>(P_{b1} \cup CA)</td>
<td>Stable models</td>
<td>(ST)</td>
<td>(ST)</td>
<td>(ST)</td>
<td>(ST)</td>
</tr>
<tr>
<td>(P_{b2} \cup CA)</td>
<td>Supported models</td>
<td>(SUPP)</td>
<td>(SUPP)</td>
<td>(SUPP)</td>
<td>(SUPP)</td>
</tr>
<tr>
<td>(P_{b3} \cup CA)</td>
<td>Supported models</td>
<td>(SUPP)</td>
<td>(SUPP)</td>
<td>(SUPP)</td>
<td>(SUPP)</td>
</tr>
<tr>
<td>(P_{b1} \cup SA)</td>
<td>New</td>
<td>(WF_{DIS})</td>
<td>(WF_{DIS})</td>
<td>(WF_{DIS})</td>
<td>(WF_{DIS})</td>
</tr>
<tr>
<td>(P_{b3} \cup SA)</td>
<td>New</td>
<td>(FIT_{DIS})</td>
<td>(FIT_{DIS})</td>
<td>(FIT_{DIS})</td>
<td>(FIT_{DIS})</td>
</tr>
<tr>
<td>(P_{b1} \cup CA)</td>
<td>New</td>
<td>(ST_{DIS})</td>
<td>(ST_{DIS})</td>
<td>(ST_{DIS})</td>
<td>(ST_{DIS})</td>
</tr>
<tr>
<td>(P_{b3} \cup CA \cup LG(P_{b4}))</td>
<td>Answer sets</td>
<td>Answer sets</td>
<td>(ANS)</td>
<td>(ANS)</td>
<td>(ANS)</td>
</tr>
</tbody>
</table>

*Note: \(ST!\) equals \(ST\) when the latter is complete; otherwise, \(ST!\) is undefined. New 
notions without a name are labeled "new." Empty boxes correspond to notions 
that have not been discussed in this paper.*
ST\textsuperscript{DIS}, ST\textsubscript{DIS}, and WF\textsubscript{DIS}, have satisfactory inferential capabilities and enjoy some interesting properties.

- \( P_{\circ,2} \) induces semantics that can recognize and ignore the rules whose body cannot be satisfied.
- The epistemic semantics allow definition of a class of applications, called ignorance tests, that cannot be satisfactorily defined by means of the old semantics for disjunctive programs.

Moreover, through the epistemic semantics, the framework led to the definition of a new notion of negation, that we called pure negation-as-failure, in contrast with the negation underlying the old semantics, which we called negation-as-assumed-falsity. We proved that the latter can be expressed as a combination of pure negation-as-failure and classical negation.

Ignorance tests are interesting per se; they provide a criterion for comparing different semantics. The criterion is based on the ability to define programs that preserve their intended meaning in different contexts. In this respect, ignorance tests are similar to the principle of stratification [36]. Ignorance tests and the principle of stratification are of practical interest. They allow a formal characterization of the semantics that allow certain applications to be implemented in a modular way.

Some of the most interesting applications of the framework had to be left for future work. We have argued that both negation-as-assumed-falsity and pure negation-as-failure are needed, and that three-valued autoepistemic logics provide an appealing formal foundation for a language with both negations. Moreover, by means of three-valued autoepistemic logics, different semantics for normal programs can be integrated, in order to achieve a better trade-off between inferential power and computational complexity. Finally, the possibility of expressing negation-as-assumed-falsity in terms of pure negation-as-failure and classical negation, provides new motivations for the study of logic programming languages with classical negation.

APPENDIX A: PROOFS

First we state some easy technical lemmas that will be applied in the proofs of the main results.

Lemma A.1 (Elementary properties of \( \models_B \)). For all belief states \( B \) for all sets of \( \mathcal{L}_L \)-sentences \( S \), and for all \( \mathcal{L}_L \)-sentences \( \psi \):

(i) If \( \psi \in B^+ \), then \( S \models_B L\psi \).
(ii) If \( \psi \in B^- \), then \( S \models_B \neg L\psi \).

Moreover, if \( S \) is \( B \)-consistent:

(iii) If \( S \models_B L\psi \), then \( \psi \in B^+ \).
(iv) If \( S \models_B \neg L\psi \), then \( \psi \in B^- \).

Proof. (i) If \( \psi \in B^+ \), then, for all \( B \)-interpretations \((I, B)\), we have \((I, B) \models B L\psi \) (by the definition of satisfaction). It follows immediately that \( S \models_B L\psi \).
(ii) Similar to the proof of (i).

(iii) We prove the contrapositive, that is, \( \psi \notin B^+ \) implies \( S \not\models_B L\psi \). By hypothesis \( S \) is \( B \)-consistent. Let \((I, B)\) be a \( B \)-model of \( S \). If \( \psi \notin B^+ \), we have \((I, B) \not\models L\psi \) and hence \( S \not\models_B L\psi \).

(iv) Similar to the proof of (iii). \[ \square \]

**Lemma A.2** (Elementary properties of \( A_S \) and \( \Theta_S \)). For all sets of \( \mathcal{L}_L \)-sentences \( S \), for all \( \mathcal{L}_L \)-sentences \( \psi \), and for all belief states \( B, B' \):

(i) \( B' \in A_S(B) \) implies \( S \models_B \Theta_S(B)^+ \).
(ii) \( S \subseteq \Theta_S(B)^+ \).
(iii) \( \psi \in B^+ \) implies \( L\psi \in \Theta_S(B)^+ \).
(iv) \( \psi \in B^- \) implies \( \neg L\psi \in \Theta_S(B)^+ \).
(v) If \( S \) is \( B \)-consistent, then \( L\psi \in \Theta_S(B)^+ \) implies \( \psi \in B^+ \).
(vi) If \( S \) is \( B \)-consistent, then \( \neg L\psi \in \Theta_S(B)^+ \) implies \( \psi \in B^- \).
(vii) If \( S \) is \( B \)-consistent, then \( \Theta_S(B)^+ \) is \( B \)-consistent.
(viii) Let \( B \) be a fixpoint of \( \Theta_S \). \( B \) is a GSE of \( S \) iff \( S \) is \( B \)-consistent.

**Proof.** (i) and (ii) follow easily from the definition of \( \Theta_S(B)^+ \).

(iii) We have \( \psi \in B^+ \):

only if, for all \( B' \supseteq B, \psi \in B'^+ \),
only if, for all \( B' \in A_S(B), \psi \in B'^+ \),
only if, for all \( B' \in A_S(B), S \models_B L\psi \),
only if \( L\psi \in \Theta_S(B)^+ \).

(iv) Similar to the proof of (iii).

(v) It suffices to show that \( \psi \in B^+ \) follows from the assumptions (a) \( S \) is \( B \)-consistent and (b) \( L\psi \in \Theta_S(B)^+ \). By (a) we have \( B \in A_S(B) \). Then (by (i)) we have \( S \models_B \Theta_S(B)^+ \) and hence, by (b), \( S \models_B L\psi \). By Lemma A.1(iii), this implies \( \psi \in B^+ \).

(vi) Similar to the proof of (v).

(vii) If \( S \) is \( B \)-consistent, then (a) \( B \in A_S(B) \) (by definition of \( A_S \)) and (b) there exists a \( B \)-model \((I, B)\) of \( S \). From (a) and (i), it follows that \( S \models_B \Theta_S(B)^+ \) and hence, by (b), \((I, B) \models \Theta_S(B)^+ \), which implies that \( \Theta_S(B)^+ \) is \( B \)-consistent.

(viii) Since \( B \) is a fixed point of \( \Theta_S \), we have that \( B \) is a GSE of \( S \) iff \( B \) is epistemically consistent. Therefore, it suffices to show that \( B \) is epistemically consistent iff \( S \) is \( B \)-consistent.

Only If: Assume that \( B \) is epistemically consistent. Then \( B \) has a \( B \)-model \((I, B)\), such that \((I, B) \models B^+ \). By (ii), we have \( S \subseteq \Theta_S(B)^+ = B^+ \) and hence \((I, B) \models S \), which proves that \( S \) is \( B \)-consistent.

If: Assume that \( S \) is \( B \)-consistent. By (vii), \( \Theta_S(B)^+ \) is \( B \)-consistent. Since \( \Theta_S(B)^+ = B^+ \) we have that \( B^+ \) is \( B \)-consistent and hence \( B \) is epistemically consistent. \[ \square \]

**Proofs for Section 2**

**Theorem 2.1.** If \( B \) is a generalized stable expansion, then \( B \) is stable, that is:
(i) $B^+ \models \psi$ implies $\psi \in B^+$.  
(ii) $L\psi \in B^+$ iff $\psi \in B^+$.  
(iii) $\neg L\psi \in B^+$ iff $\psi \in B^-$.  

**Proof.** Suppose that $B$ is a GSE of $S$ and note that this implies $B = \Theta_S(B)$.  
(i) We have $B^+ \models \psi$:  
  iff, for all belief states $B', B^+ \models_{B'} \psi$,  
  only if, for all $B' \in A_S(B)$, $B^+ \models_{B'} \psi$,  
  only if, for all $B' \in A_S(B)$, $\Theta_S(B)^+ \models_{B'} \psi$ (because $B = \Theta_S(B)$),  
  only if, for all $B' \in A_S(B)$, $S \models_{B'} \psi$, (by Lemma A.2(i)),  
  only if, $\psi \in \Theta_S(B)^+ = B^+$.  

(ii) $B$ is a GSE of $S$ and a fixed point of $\Theta_S$. Thus, by Lemma A.2 (viii), we have that $S$ is $B$-consistent. By Lemma A.2(iii, v), this implies that $L\psi \in \Theta_S(B)^+$ holds iff $\psi \in B^+$. Then (ii) immediately follows.  
(iii) Similar to the proof of (ii). \qed  

**Lemma 2.1 (Monotonicity).** $B_1 \subseteq B_2$ implies $\Theta_S(B_1) \subseteq \Theta_S(B_2)$.  

**Proof.** Assume that $B_1 \subseteq B_2$. We have to show that $\Theta_S(B_1) \subseteq \Theta_S(B_2)$. By assumption we have  
$$A_S(B_1) = \{ B' \mid B' \text{ is a belief state and } B' \supseteq B_1 \text{ and } S \text{ is } B\text{-consistent} \}$$  
$$\supseteq \{ B' \mid B' \text{ is a belief state and } B' \supseteq B_2 \text{ and } S \text{ is } B\text{-consistent} \}$$  
$$= A_S(B_2).$$  

From the above disequality and the definition of $\Theta_S$ it follows easily that $\Theta_S(B_1) \subseteq \Theta_S(B_2)$. \qed  

**Lemma A.3 (Elementary properties of complete belief states).** If $B$ is a complete belief state, then, for all sets of $\mathcal{L}_L$-sentences $S$ and all $\mathcal{L}_L$-sentences $\phi$:  
(i) $S \models_B \phi$ holds iff $S \cup LB^+ \cup \neg LB^+ \models \phi$, where  
$$LB^+ = \{ L\sigma \mid \sigma \in B^+ \},$$  
$$\neg LB^+ = \{ \neg L\sigma \mid \sigma \in B^+ \}.$$  

Moreover, if $S$ is $B$-consistent:  
(ii) $A_S(B) = \{ B \}$,  
(iii) $\Theta_S(B)^+ = \{ \phi \mid S \cup LB^+ \cup \neg LB^+ \models \phi \}$,  
(iv) $\Theta_S(B)^- = \{ \phi \mid S \cup LB^+ \cup \neg LB^+ \not\models \phi \}$.  

**Proof.** (i) For all belief states $B'$ we have $(I, B') \models LB^+ \cup \neg LB^+$:  
  iff $(I, B') \models L\sigma \mid \sigma \in B^+$ \cup $\neg L\sigma \mid \sigma \in B^+$,  
  iff $(I, B') \models L\sigma \mid \sigma \in B^+$ \cup $\neg L\sigma \mid \sigma \in B^-$ (by completeness of $B$),  
  iff $B' \supseteq B$,  
  iff $B' = B$ (because complete belief states are maximal).
In other words, the set of models of $LB^+ \cup \neg LB^+$ is precisely the set of $B$-interpretations. It follows that:

(a) $S \models_B \phi$ is equivalent to $S \cup LB^+ \cup \neg LB^+ \models \phi$.

It follows also that all the models of $LB^+ \cup \neg LB^+$ are complete, and hence:

(b) $S \cup LB^+ \cup \neg LB^+ \models \phi$ holds iff every complete model of $S \cup LB^+ \cup \neg LB^+$ satisfies $\phi$.

Note that complete models are in one-to-one correspondence with classical models (i.e., classical truth assignments to ordinary and autoepistemic atoms). By (b), this implies that:

(c) $S \cup LB^+ \cup \neg LB^+ \models \phi$ holds iff $S \cup LB^+ \cup \neg LB^+ \models \phi$.

From (a) and (c) we immediately get (i).

(ii) Complete belief states are maximal belief states, and hence

$$\mathcal{A}_S(B) \subseteq \{B' \mid B' \text{ is a belief state and } B' \supseteq B\} = \{B\}.$$  

We are left to show that $\mathcal{A}_S(B) \supseteq \{B\}$, which follows immediately from the assumption that $S$ is $B$-consistent.

(iii) and (iv) follow immediately from (i) and (ii) and the definition of $\Theta_S$. 

Theorem 2.2. For all sets of $L_L$-sentences $S$ and $T$, $T$ is a consistent stable expansion of $S$ iff $(T, T)$ is a complete GSE of $S$.

Proof. If: If $(T, \overline{T})$ is a complete GSE of $S$, then $(T, \overline{T})$ is a fixed point of $\Theta_S$. From this fact and Lemma A.3(iii), we have

$$T = \langle T, \overline{T} \rangle^+ = \Theta_S((T, \overline{T}))^+ = \{\psi \mid S \cup LT \cup \neg LT \models \psi\}.$$

This proves that $T$ is a stable expansion of $S$. Moreover, since GSE’s are epistemically consistent, $(T, \overline{T})$ must have an autoepistemic model, which implies that $T$ is consistent.

Only if: Assume that $T$ is a consistent stable expansion of $S$. We have to show that $(T, \overline{T})$ is an epistemically consistent fixed point of $\Theta_S$. By the definition of stable expansion we have

$$T = \{\psi \mid S \cup LT \cup \neg LT \models \psi\}.$$

It follows immediately, by Lemma A.3(iii, iv), that $(T, \overline{T})$ is a fixed point of $\Theta_S$. We are left to show that $(T, \overline{T})$ is epistemically consistent. By assumption, $T$ has a model, that will be denoted by $(I, B)$. Note that $T$ contains $LT \cup \neg LT$; therefore, $(I, B)$ must satisfy $LT \cup \neg LT$. This implies $B = \langle T, \overline{T} \rangle$ (see the proof of Lemma A.3(i)). It follows immediately that $(I, B)$ is an autoepistemic model of $(T, \overline{T})$, i.e., $(T, \overline{T})$, is epistemically consistent. 

Lemma A.4. The following are equivalent:

(i) $B$ is a complete GSE of $S$.
(ii) $B$ is a GSE of $S \cup CA$. 
(iii) $B$ is a minimal GSE of $S \cup CA$.

**Proof.** (i) $\iff$ (ii): The reader may easily verify that:

(a) $S \cup CA$ is $B$-consistent iff $S$ is $B$-consistent and $B$ is complete. Then derive:

(b) If $S \cup CA$ is $B$-consistent, then $\Theta_{S\cup CA}(B) = \Theta_S(B)$. In order to prove this, assume that $S \cup CA$ is $B$-consistent. By (a), we have that $S$ is $B$-consistent and $B$ is complete. Then Lemma A.3(iii, iv) can be applied to $S \cup CA$, $S$, and $B$, and we get

$$\Theta_{S\cup CA}(B)^+ = \{\phi \mid S \cup CA \cup LB^+ \cup \neg LB^+ \vdash \phi\}$$

by Lemma A.3 (iii)

$$= \{\phi \mid S \cup LB^+ \cup \neg LB^+ \vdash \phi\}$$

because $CA$ is a set of tautologies

$$= \Theta_S(B)^+$$

by Lemma A.3(iii).

Similarly, derive $\Theta_{S\cup CA}(B)^- = \Theta_S(B)^-$. It follows that $\Theta_{S\cup CA}(B) = \Theta_S(B)$.

(c) If $B$ is a GSE of $S \cup CA$, then $B$ is a complete GSE of $S$. In order to prove this, assume that $B$ is a GSE of $S \cup CA$. Then $S \cup CA$ is $B$-consistent (by Lemma A.2(viii)). By (a), we have that $B$ is complete, so we are left to show that $B$ is a GSE of $S$. Since $S \cup CA$ is $B$ consistent, (b) implies that $\Theta_{S\cup CA}(B) = \Theta_S(B)$. From this equality and the assumption, it follows that $B$ must be an epistemically consistent fixed point of $\Theta_S(B)$, hence, a GSE of $S$.

(d) If $B$ is a complete GSE of $S$, then $B$ is a GSE of $S \cup CA$. In fact, if $B$ is a complete GSE of $S$, then $S$ is $B$-consistent (by Lemma A.2(viii)) and $B$ is complete, and hence, by (a), we have that $S \cup CA$ is $B$-consistent. By (b), this implies that $\Theta_{S\cup CA}(B) = \Theta_S(B)$. Since $B$ must be an epistemically consistent fixed point of $\Theta_S(B)$, it follows that $B$ must be an epistemically consistent fixed point of $\Theta_{S\cup CA}(B)$, hence, a GSE of $S \cup CA$.

From (c) and (d), conclude that (i) and (ii) are equivalent.

(ii) $\iff$ (iii): Since all the GSE's of $S \cup CA$ are complete, and complete belief states are maximal belief states, the GSE’s of $S \cup CA$ are never comparable with each other. It follows that they are all minimal GSE’s. Consequently, (ii) and (iii) are equivalent. $\Box$

**Theorem 2.3.** If $T$ is a consistent stable expansion of $S$, then $(T, \overline{T})$ is a GSE of $S \cup CA$. Conversely, if $B$ is a GSE of $S \cup CA$, then $B^+$ is a consistent stable expansion of $S$ and $B^- = \overline{T}$.

**Proof.** It follows immediately from Theorem 2.2 and Lemma A.4(i, ii). $\Box$

**Theorem 2.4.** Let $S$ be an implicative set of sentences and let $B = lfp(\Theta_S)$. Then the following are equivalent:

(i) $S$ has a GSE.

(ii) $B^+$ is consistent.

(iii) $B$ is the least GSE of $S$. 


and $\text{Cons}(S, B')$ is consistent, then let $I$ be a model of $\text{Cons}(S, B')$. By (ii), $(I, B')$ is a model of $S$ and hence, $S$ is $B'$-consistent. It follows immediately that $B' \in \mathcal{A}_S(B)$.

(iv) We have $S \models_B \varphi$:

$$
\text{iff, for all } B\text{-models } (I, B) \text{ of } S: (I, B) \models \varphi,
$$

$$
\text{iff, for all } B\text{-models } (I, B) \text{ of } S: I \models \varphi,
$$

$$
\text{iff, for all (classical) models } I \text{ of } \text{Cons}(S, B)I \models \varphi \text{ (by (ii))},
$$

$$
\text{iff } \text{Cons}(S, B) \vdash \varphi.
$$

(v) We prove that $\varphi \notin \Theta_S(B)^+$ iff $\text{Cons}(S, B) \not\models \varphi$ which is equivalent. We have $\varphi \notin \Theta_S(B)^+$:

$$
\text{iff, for some } B' \in \mathcal{A}_S(B), S \not\models_{B'} \varphi,
$$

$$
\text{iff, for some } B' \in \mathcal{A}_S(B), \text{Cons}(S, B') \not\models \varphi,
$$

$$
\text{iff } \text{Cons}(S, B) \not\models \varphi. \text{ (Proof of the last equivalence: assume that, for some } B' \in \mathcal{A}_S(B), \text{Cons}(S, B') \not\models \varphi. \text{ Note that } B' \supseteq B \text{ and hence, by (i), } \text{Cons}(S, B) \subseteq \text{Cons}(S, B'). \text{ It follows, by monotonicity of classical logic, that } \text{Cons}(S, B) \not\models \varphi. \text{ Conversely, if } \text{Cons}(S, B) \not\models \varphi, \text{ then let } B' = B. \text{ Obviously, we have } B' \supseteq B \text{ and } \text{Cons}(S, B') \not\models \varphi \text{ and hence, by (iii), } B' \in \mathcal{A}_S(B).)
$$

(vi) We prove that $\varphi \notin \Theta(B)^-$ iff, for some $B' \supseteq B$, $\text{Cons}(S, B')$ is consistent and $\text{Cons}(S, B') \vdash \varphi$. We have $\varphi \notin \Theta(B)^-$:

$$
\text{iff, for some } B' \in \mathcal{A}_S(B), S \models_{B'} \varphi,
$$

$$
\text{iff, for some } B' \in \mathcal{A}_S(B), \text{Cons}(S, B') \vdash \varphi \text{ (by (iv))},
$$

$$
\text{iff, for some } B' \supseteq B, \text{Cons}(S, B') \text{ is consistent and } \text{Cons}(S, B') \vdash \varphi \text{ (by (iii))}.
$$

(vii) $\varphi \in B^+$:

$$
\text{iff } \varphi \in \Theta_S(B)^+ \text{ (because } B = \Theta_S(B)),
$$

$$
\text{iff } \text{Cons}(S, B) \vdash \varphi \text{ (by (v))}.
$$

(viii) It follows immediately from (vi). \qed

Lemma A.5 (Elementary properties of quasiprograms). Let $S$ be a quasiprogram. For all belief states $B, B'$, for all $\mathcal{L}_L$-sentences $\varphi$, and for all ordinary sentences $\varphi$:

(i) $S$ is $B$-consistent.
(ii) $B' \in \mathcal{A}_S(B)$ iff $B' \supseteq B$.
(iii) $\varphi \in \Theta_S(B)^-$ iff, for all $B' \supseteq B$, $\text{Cons}(S, B') \not\models \varphi$.
(iv) $L\phi \in \Theta_S(B)^+$ iff $\phi \in B^+$.
(v) $\neg L\phi \in \Theta_S(B)^+$ iff $\phi \in B^-$.
(vi) $L\phi \in \Theta_S(B)^-$ iff $\phi \in B^-$.
(vii) $\neg L\phi \in \Theta_S(B)^-$ iff $\phi \in B^+$.

Proof. (i) By definition of quasiprograms, $\text{Heads}(S)$ has a (classical) model $I$. Note that, for all $B$, $(I, B)$ is a model of $\text{Heads}(S)$, hence, a model of $S$. It follows that $S$ is always $B$-consistent.

(ii) It follows immediately from (i) and the definition of $\mathcal{A}_S$.

(iii) We have $\varphi \in \Theta_S(B)^-$:
PROOF. We will prove the following implications: (i) ⇒ (ii), (ii) ⇒ (iii), (iii) ⇒ (i).

(i) ⇒ (ii): Assume that \( S \) has a GSE \( B' \). \( B' \) must be epistemically consistent; consequently, \( B'^+ \) must be consistent. Moreover, \( B' \) must be fixed point of \( \Theta_S \). Then \( B = \text{fp}(\Theta_S) \subseteq B' \) and hence, \( B^+ \subseteq B'^+ \). Since \( B'^+ \) is consistent, it follows immediately that \( B^+ \) is consistent, too.

(ii) ⇒ (iii): Assume that \( B^+ \) is consistent and let \((I, B')\) be a model of \( B^+ \). We have:

- (a) \( I \models Cons(S, B) \). In fact, Lemma 2.2(vii) implies \( B^+ \supseteq Cons(S, B) \). From the assumption \((I, B') \models B^+ \) derive \((I, B') \models Cons(S, B) \) and hence, \( I \models Cons(S, B) \) (because the interpretation of ordinary sentences, in \((I, B')\), is determined by \( I \)).
- (b) \( S \) is \( B \)-consistent. In fact, \((I, B) \models S \) (by (a) and Lemma 2.2(ii)).
- (c) \( B \) is a GSE of \( S \) (by (b) and Lemma 2.2(viii)).

Finally note that \( B \), being the least fixed point of \( \Theta_S \), is necessarily the least GSE of \( S \).

(iii) ⇒ (i): Trivial. □

Lemma 2.2 (Elementary properties of implicative theories). If \( S \) is an implicative set of sentences and \( \varphi \) is an ordinary sentence, then:

(i) \( B \subseteq B' \) implies \( Cons(S, B) \subseteq Cons(S, B') \).
(ii) \( (I, B) \models S \) iff \( I \models Cons(S, B) \).
(iii) \( B' \in \mathcal{A}_S(B) \) iff \( B' \supseteq B \) and \( Cons(S, B') \) is consistent.
(iv) \( S \models_B \varphi \) iff \( Cons(S, B) \models \varphi \).
(v) \( \varphi \in \Theta_S(B)^+ \) iff \( Cons(S, B) \models \varphi \).
(vi) \( \varphi \in \Theta_S(B)^- \) iff for all \( B' \supseteq B \), either \( Cons(S, B') \) is inconsistent or \( Cons(S, B') \models \varphi \).

Moreover, if \( B \) is a fixed point of \( \Theta_S \):

(vii) \( \varphi \in B^+ \) iff \( Cons(S, B) \models \varphi \).
(viii) \( \varphi \in B^- \) iff for all \( B' \supseteq B \), either \( Cons(S, B') \) is inconsistent or \( Cons(S, B') \models \varphi \).

PROOF. (i) The reader may easily verify that Kleene's valuation is monotonic with respect to classical connectives, that is, if \( B \subseteq B' \), \( \sigma \) does not contain \( \leftrightarrow \), and \( \sigma \) is true (resp. false) in \((I, B)\), then \( \sigma \) is true (resp. false) in \((I, B')\). It follows that \( (\varphi \leftarrow \sigma) \in Active(S, B) \) only if \( (\varphi \leftarrow \sigma) \in Active(S, B') \), that is, \( Active(S, B) \subseteq Active(S, B') \). Conclude that \( Cons(S, B) \subseteq Cons(S, B') \).

(ii) We prove that \((I, B) \not\models S \) iff \( I \not\models Cons(S, B) \), which is equivalent. We have \((I, B) \not\models S \):

iff for some rule \( (\varphi \leftarrow \sigma) \in S : (I, B) \not\models (\varphi \leftarrow \sigma) \),
iff for some rule \( (\varphi \leftarrow \sigma) \in S : (I, B) \models \sigma \) and \((I, B) \not\models \varphi \),
iff for some rule \( (\varphi \leftarrow \sigma) \in S, (\varphi \leftarrow \sigma) \in Active(S, B) \) and \( I \not\models \varphi \),
iff \( \not\models Cons(S, B) \).

(iii) If \( B' \in \mathcal{A}_S(B) \), then \( B' \supseteq B \) and there is a model \((I, B')\) of \( S \). By (ii), we have \( I \models Cons(S, B') \) and hence \( Cons(S, B') \) is consistent. Conversely, if \( B' \supseteq B \)
iff, for all $B' \in A_\mathcal{S}(B), S \not\models_{B'} \varphi$ by definition of $\Theta_\mathcal{S}$,
iff, for all $B' \supseteq B, S \not\models_{B'} \varphi$, by (ii),
iff, for all $B' \supseteq B, \text{Cons}(S, B') \not\models \varphi$ by Lemma 2.2(iv).

(iv) By (i), $S$ is $B$-consistent. By Lemma A.2(iii, v), this implies that $L\phi \in \Theta_\mathcal{S}(B)$ iff $\phi \in B^+$.
(v) Similar to the proof of (iv).
(vi) We have $L\phi \in \Theta_\mathcal{S}(B)^-$:
iff, for all $B' \in A_\mathcal{S}(B), S \not\models_{B'} L\phi$.
iff, for all $B' \in A_\mathcal{S}(B), \phi \notin B'^+$ by Lemma A.2(iii, v),
iff, for all belief states $B' \supseteq B, \phi \notin B'^+$ by (ii),
iff $\phi \in B^-$.

(vii) Similar to the proof of (vi). □

**Theorem 2.5.** If $S$ is a quasiprogram, then $\text{lfp}(\Theta_S)$ is the least GSE of $S$.

**PROOF.** Let $B = \text{lfp}(\Theta_S)$. By Lemma A.5(i), $S$ is $B$-consistent. Then, by Lemma A.2(viii), $B$ is a GSE of $S$ and hence, by Theorem 2.4, $B$ is the least GSE of $S$. □

**Proofs for Section 4**

**Relations between $\Theta_{P_{b1}}, \Pi_P$, and $\Phi_P$**

**Lemma 4.1** For all complete belief states $B$,

\[ \Pi_P(\text{Atom}(B^+)) = \text{Atom}(\Theta_{P_{b1}}(B)^+) . \]

**PROOF.** We have

\[
\Pi_P(\text{Atom}(B^+)) = -[\Psi_P(\text{Atom}(B))^-] \quad \text{by definition of } \Phi_P
\]
\[
= -[\text{Atom}(\Theta_{P_{b1}}(B))^-] \quad \text{by Lemma 4.2}
\]
\[
= \text{Atom}(\Theta_{P_{b1}}(B)^-)
\]
\[
= \text{Atom}(\Theta_{P_{b1}}(B)^+) \quad \text{because } B \text{ is complete}.
\]

□

**Lemma 4.2.** For all belief states $B$, $\Psi_P(\text{Atom}(B)) = \text{Atom}(\Theta_{P_{b1}}(B))$.

In order to prove this lemma, note that $P_{b1}$ is equivalent to the quasiprogram $U$, obtained from $P_{b1}$ by replacing each implication $(a \leftarrow b_1 \land \cdots \land b_n \land \neg L_{c1} \land \cdots \land \neg L_{cm})$ with the equivalent implication

\[ ((a \leftarrow b_1 \land \cdots \land b_n) \leftarrow \neg L_{c1} \land \cdots \land \neg L_{cm}) . \]

$U$ enjoys the following property:

**Lemma A.6.** For all belief states $B$, $\text{Cons}(U, B) = P^I$, where $I = \text{Atom}(\overline{B^-})$. 

PROOF. We have

\[
\text{Cons}(U, B) = \text{Ord}(U) \cup \{ \varphi \mid (\varphi \leftrightarrow \sigma) \in U \text{ and } \models_B \sigma \}
\]

\[
= \text{Ord}(U) \cup \{ a \leftarrow b_1 \land \cdots \land b_n \mid (a \leftarrow b_1 \land \cdots \land b_n) \leftarrow \neg Lc_1 \land \cdots \land \neg Lc_m \in U \land c_1, \ldots, c_m \in B^- \}
\]

\[
= \{ a \mid a \in \text{Atom}(P) \} \cup \{ a \leftarrow b_1 \land \cdots \land b_n \mid (a \leftarrow b_1 \land \cdots \land b_n \land \sim c_1 \land \cdots \land \sim c_m) \in P \land c_1, \ldots, c_m \notin B^- \}
\]

\[
= P^I, \quad \text{where } I = \text{Atom}(B^-).
\]

\[
\]

PROOF OF LEMMA 4.2. Note that \( \Theta_{P_{b1}} = \Theta_U \), where \( U \) is the quasiprogram corresponding to \( P_{b1} \). Thus, it suffices to show that

\[
\text{Atom}(\Theta_U(B)) = \Psi_P(\text{Atom}(B)) = (\Pi_P(I), -\Pi_P(J)),
\]

where

\[
I = -\text{Atom}(B^-) = \text{Atom}(-B^-),
\]

\[
J = \text{Atom}(B^+) = \text{Atom}(B^+).
\]

(a) For all atoms \( a, a \in \Theta_U(B)^+ \) iff \( a \in \Pi_P(I) \). In fact, we have \( a \in \Theta_U(B)^+ \):

\[
\text{iff } \text{Cons}(U, B) \vdash a \text{ by Lemma 2.2(v)},
\]

\[
\text{iff } P^I \vdash a \text{ by Lemma A.6(ii)},
\]

\[
\text{iff } a \in \text{lm}(P^I),
\]

\[
\text{iff } a \in \Pi_P(I).
\]

(b) \( \Pi_P(J) = \text{max}\Sigma \), where \( \Sigma = \{ \Pi_P(I') \mid I' = \text{Atom}(B'^-), \text{and } B' \supseteq B \} \). For all belief states \( B' \supseteq B \) we have \( B'^- \supseteq B^+ \) and hence, \( I' = \text{Atom}(B'^-) \supseteq \text{Atom}(B^+) = J \). Since \( \Pi_P \) is antimonotonic (see [4]), this implies \( \Pi_P(I') \subseteq \Pi_P(J) \). This shows that \( \Pi_P(J) \) is an upper bound for \( \Sigma \). We are left to show that \( \Pi_P(J) \) is also a member of \( \Sigma \). In order to prove this, let \( B' = (B^+, B^+) \). Obviously, \( B' \supseteq B \) and hence, \( \Pi_P(I') \in \Sigma \). Moreover, \( I' = \text{Atom}(B'^-) = J \). It follows immediately that \( \Pi_P(J) \in \Sigma \).

(c) For all atoms \( a, a \in \Theta_U(B)^- \) iff \( a \notin \Pi_P(J) \). In fact, we have \( a \in \Theta_U(B)^- \):

\[
\text{iff, for all } B' \supseteq B, \text{Cons}(U, B') \not\vdash a,
\]

\[
\text{iff, for all } B' \supseteq B, P^I \not\vdash a, \text{ where } I' = \text{Atom}(B'^-),
\]

\[
\text{iff, for all } B' \supseteq B, a \notin \text{lm}(I'), \text{ where } I' = \text{Atom}(B'^-),
\]

\[
\text{iff, for all } B' \supseteq B, a \notin \Pi_P(I'), \text{ where } I' = \text{Atom}(B'^-),
\]

\[
\text{iff, for all } \Pi_P(I') \subseteq \Sigma, a \notin \Pi_P(I'), \text{ where } I' = \text{Atom}(B'^-),
\]

\[
\text{iff, a } \notin \Pi_P(J) \text{ by (b)}.
\]

By (a) and (c) we have \( \text{Atom}(\Theta_U(B)) = (\Pi_P(I), -\Pi_P(J)) \). □

Relations between \( \Theta_{P_{b2}}, \Theta_{P_{b3}}, \) and Fitting's Operator

Lemma A.7. Let \( \sigma = L\phi_1 \land \cdots \land L\phi_n \land \neg L\psi_1 \land \cdots \land \neg L\psi_m \), let \( i \) range over \( 1, \ldots, n \), and let \( j \) range over \( 1, \ldots, m \). For all belief states \( B \), the following are equivalent:
(i) For all $B' \supseteq B$, $\not\equiv_{B'} \sigma$.
(ii) Some of the following conditions hold:
  - For some $i$ and $j$, $\phi_i = \psi_j$.
  - For some $i$, $\phi_i \in B^-$.
  - For some $j$, $\psi_j \in B^+$.

**Proof.** There are two possibilities:

(a) For some $i$ and $j$, $\phi_i = \psi_j$.
(b) For all $i$ and $j$, $\phi_i \neq \psi_j$.

If (a) holds, then $\sigma$ is obviously unsatisfiable (it contains the complementary literals $L\phi_i$ and $\neg L\psi_j = \neg L\phi_i$). It follows immediately that (i) and (ii) are both true.

We are left to show that (i) and (ii) are equivalent under the assumption that (b) holds. We have:

(c) If (ii) holds, then (i) holds. In order to prove this, assume that (ii) holds. Since we assumed (b), there are only two possibilities:
  - For some $i$, $\phi_i \in B^-$.
  - For some $j$, $\psi_j \in B^+$.

In the first case, for all belief states $B' \supseteq B$, we have $\phi_i \in B'^-$, which implies $\models_{B'} \neg L\phi_i$ and hence $\not\equiv_{B'} \sigma$. Similarly, in the second case, we have that for all belief states $B' \supseteq B$, $\psi_j \in B'^+$, which implies $\models_{B'} L\psi_j$ and hence $\not\equiv_{B'} \sigma$.

Conclude that (i) holds.

(d) If (ii) does not hold, then (i) does not hold. In order to prove this, assume that (ii) does not hold, that is, for all $i$ and $j$, $\phi \notin B^-$ and $\psi_j \notin B^-$. It follows that $(B^+ \cup \{\phi_1, \ldots, \phi_n\}) \cap (B^- \cup \{\psi_1, \ldots, \psi_n\}) = \emptyset$, and hence $B' = \langle B^+ \cup \{\phi_1, \ldots, \phi_n\}, B^- \cup \{\psi_1, \ldots, \psi_n\} \rangle$ is a belief state. Obviously, we have $B' \supseteq B$ and $\models_{B'} \sigma$, that is, (i) does not hold.

Thus, by (c) and (d), (i) and (ii) are equivalent. $\square$

**Lemma A.8.** Let $P$ be any program and let $P'$ be the program obtained from $P$ by eliminating all the clauses that contain a pair of complementary literals in the body. For all belief states $B$,

$$\Phi_{P'}(\text{Atom}(B)) = \text{Atom}(\Theta_{P_2}(B)).$$

**Proof.** For all atoms $a$, we have $a \in \Theta_{P_2}(B)^+$:

- iff $\text{Cons}(P_{\phi_2}, B) \vdash a$ by Lemma 2.2 (v),
- iff $a \in \text{Cons}(P_{\phi_2}, B)$ because $\text{Cons}(P_{\phi_2}, B)$ is a set of atoms,
- iff either $a \in P_{\phi_2}$ or there exists $(a \leftarrow \sigma) \in P_{\phi_2}$ s.t. $\models_{B} \sigma$ by definition of $\text{Cons}$,
- iff either $a \in P_{\phi_2}$ or there exists $(a \leftarrow b_1 \land \cdots \land b_n \land \neg c_1 \land \cdots \land \neg c_m) \in P_{\phi_2}$ s.t. $b_1, \ldots, b_n \in B^+$ and $c_1, \ldots, c_m \in B^-$,
- iff either $a \in P$ or there exists $(a \leftarrow b_1 \land \cdots \land b_n \land c_1 \land \cdots \land c_m) \in P$ s.t. $b_1, \ldots, b_n \in B^+$ and $c_1, \ldots, c_m \in B^-$. 


iff there is a \( P \) or there exists \((a \leftarrow b_1 \wedge \cdots \wedge b_n \wedge \sim c_1 \wedge \cdots \wedge \sim c_m) \in P'\) s.t. 
b_1, \ldots, b_n \in B^+ and \(c_1, \ldots, c_m \in B^-\), (by noting that \(b_1, \ldots, b_n \in B^+\) and \(c_1, \ldots, c_m \in B^-\) imply \(b_i \neq c_j\), for all \(i, j\), hence \((a \leftarrow b_1 \wedge \cdots \wedge b_n \wedge \sim c_1 \wedge \cdots \wedge \sim c_m) \in P'\)),
iff \( a \in \Phi_{P'}(\text{Atom}(B))^+ \),

which shows that \( \Phi_{P'}(\text{Atom}(B))^+ = \text{Atom}(\Theta_{P_2}(B))^+ \). Similarly, we have \( a \in \Theta_{P_2}(B)^- \):

iff for all \( B' \supseteq B \), \( \text{Cons}(P_{P_2}, B') \neq a \) by Lemma A.5(iii),
iff for all \( B' \supseteq B \), \( a \not\in \text{Cons}(P_{P_2}, B') \) because \( \text{Cons}(P_{P_2}, B') \) is a set of atoms,
iff for all \( B' \supseteq B \), \( a \not\in P_{P_2} \) and, for all rules \((a \leftarrow \sigma) \in P_{P_2}, \not\in B', \sigma\),
iff a \( \not\in P_{P_2} \) and, for all rules \((a \leftarrow \sigma) \in P_{P_2}, \text{and all } B' \supseteq B, \text{ and } B' \not\in B', \sigma\),
iff for all \( B' \supseteq B \), \( a \not\in P' \) and, for all rules \((a \leftarrow \sigma) \in P_{P_2}, \text{ and all } B' \supseteq B, \text{ and } B' \not\in B', \sigma\),

some of the following conditions hold:

- for some \(i \) and \(j\), \(b_i = c_j\),
- for some \(i\), \(b_i \in B^-\),
- for some \(j\), \(c_j \in B^+\) (by Lemma A.7),

iff a \( \not\in P' \) and, for all rules \((a \leftarrow b_1 \wedge \cdots \wedge b_n \wedge \sim c_1 \wedge \cdots \wedge \sim c_m) \in P'\) some of the following conditions hold:

- for some \(i\), \(b_i \in B^-\),
- for some \(j\), \(c_j \in B^+\),

iff \( a \in \Phi_{P'}(\text{Atom}(B))^+ \).

This proves that \( \Phi_{P'}(\text{Atom}(B))^+ = \text{Atom}(\Theta_{P_2}(B))^+ \), which completes the proof. \( \square \)

**Theorem 4.5.** Let \( P \) be any program and let \( P' \) be the program obtained from \( P \) by eliminating all the clauses that contain a pair of complementary literals in the body. Finally, let \( B \) be the least GSE of \( P_{P_2} \). Then

\[ \text{Atom}(B) = \text{FIT}(P'). \]

**Proof.** It follows easily from Lemma A.8. \( \square \)

**Lemma 4.4.** For all belief states \( B \), \( \Phi_P(\text{Atom}(B)) = \text{Atom}(\Theta_{P_3}^2(B)). \)

**Proof.** For all atoms \( a \), we have \( a \in \Theta_{P_3}^2(B)^+ \):

iff either \( a \in P_{P_3} \) or there exists \((a \leftarrow \sigma) \in P_{P_3}, \text{ s.t. } Lb_1, \ldots, Lb_n \in \Theta_{P_3}(B)^+ \) and \( \sim c_1, \ldots, \sim c_m \in \Theta_{P_3}(B)^+ \) (see the proof of Lemma A.8),
iff either \( a \in P_{P_3} \) or there exists \((a \leftarrow \sigma) \in P_{P_3}, \text{ s.t. } b_1, \ldots, b_n \in B^+ \) and \( c_1, \ldots, c_m \in B^- \) (by Lemma A.5(iv, v)),
iff either \( a \in P \) or there exists \((a \leftarrow \sigma) \in P, \text{ s.t. } b_1, \ldots, b_n \in B^+ \) and \( c_1, \ldots, c_m \in B^- \),
iff \( a \in \Phi_P(\text{Atom}(B))^+ \).
This proves that $\Phi_P(\text{Atom}(B))^+ = \text{Atom}(\Theta^2_{P_{b3}}(B))^+$. We are left to show that $\Phi_P(\text{Atom}(B))^+ = \text{Atom}(\Phi^2_{P_{b3}}(B))^+$. We have $a \in \Theta_{P_{b3}}(B)^-$:

iff $a \notin P_{b3}$ and, for all rules $(a \leftarrow LLb_1 \land \cdots \land LLb_n \land L\neg Lc_1 \land \cdots \land L\neg Lc_m) \in P_{b3}$ the following condition holds: for some $i$, $Lb_i \in \Theta_{P_{b3}}(B)^-$ or, for some $j$, $\neg Lc_j \in \Theta_{P_{b3}}(B)^-$ (see Lemma A.8),

iff $a \notin P$ and, for all rules $(a \leftarrow b_1 \land \cdots \land b_n \land \neg c_1 \land \cdots \land \neg c_m) \in P$ some of the following conditions hold:

- for some $i, b_i \in B^-$
- for some $j, c_j \in B^+$ (by Lemma A.5(vi, vii)),

iff $a \in \Phi_P(\text{Atom}(B))^-$.

This proves that $\Phi_P(\text{Atom}(B))^+ = \text{Atom}(\Theta_{P_{b3}}(B))^-$, which completes the proof. □

Proofs for Section 5
First note that $\text{Cons}$ is compositional, when applied to $P_{bB} \cup SA$, that is:

**Proposition A.1.** For all $B$ and $B'$,

$$\text{Cons}(P_{bB} \cup SA, B \cup B') = P_{bB} \cup \text{Cons}(SA, B) \cup \text{Cons}(SA, B').$$

**PROOF.** Straightforward from the definitions of $P_{bB}$ and SA. □

Then we need a technical lemma that will be helpful for many of the following results.

**Lemma A.9.** Let $S$ be a P-state, let $I$ be a minimal model of $P_{bB} \cup S$, and let $I'$ be a (not necessarily minimal) model of $P_{bB}$. If $I' \subseteq I$, then

$$(I' \cap \mathcal{H}) = (I \cap \mathcal{H}).$$

**PROOF.** Suppose not. Then, for some minimal model $I$ of $P_{bB} \cup S$, there exists a model $I'$ of $P_{bB}$ such that $I' \subseteq I$ and $(I' \cap \mathcal{H}) \neq (I \cap \mathcal{H})$. Let

$$I'' = (I' \cap \mathcal{H}) \cup (I \cap \mathcal{H}).$$

Note that $I' \subseteq I'' \subseteq I$. We have:

(a) $I'' \models P_{bB}$. In fact, for all rules $R$ of $P_{bB}$ we have $I' \models R$. Then there are three possibilities:

(a.1) $I' \models a$, for some $a$ occurring in the head of $R$.
(a.2) $I' \models B_a$, for some $\neg B_a$ occurring in the body of $R$.
(a.3) $I' \not\models b$, for some $b \in \mathcal{H}$ occurring in the body of $R$.

In each case we have $I'' \models R$. In fact:
(a.4) From (a.1) and \( I' \subseteq I'' \) we have \( I'' \models a \), hence \( I'' \models R \).

(a.5) From (a.2) and \( I' \subseteq I'' \) we have \( I'' \models B_a \), hence \( I'' \models R \).

(a.6) From the definition of \( I'' \) we have \( (I'' \cap \mathcal{H}) = (I' \cap \mathcal{H}) \). From this fact and (a.3) conclude \( I'' \not\models b \), hence \( I'' \models R \).

This is true for every rule of \( P_{\gamma B} \); therefore, \( I'' \models P_{\gamma B} \).

(b) \( I'' \models S \). In fact, the atoms of \( \mathcal{H}' \) are given the same truth value by \( I'' \) and \( I \), and hence the assumption \( I \models S \) implies \( I'' \models S \).

Thus \( I'' \) is a model of \( P_{\gamma B} \cup S \). It follows that \( I'' = I \), because \( I \) is a minimal model of \( P_{\gamma B} \cup S \). From the definition of \( I'' \) and from the above equality conclude \( (I' \cap \mathcal{H}) = (I'' \cap \mathcal{H}) = (I \cap \mathcal{H}) \). This contradicts the assumption \( (I' \cap \mathcal{H}) \neq (I \cap \mathcal{H}) \).

Now we prove that the least GSE of \( P_{\gamma B} \cup S_\delta \) is a correct and complete representation of the stationary semantics. In the following \( \Theta_{P_{\gamma B} \cup S_\delta} \) will be abbreviated by \( \Theta \), and \( (\Theta_{P_{\gamma B} \cup S_\delta} \uparrow \alpha) \) will be abbreviated by \( \Theta_{\alpha} \).

**Lemma A.10** (Correctness). Let \( S_\delta \) be the stationary state of \( P \). For all ordinals \( \alpha \):

1. \( p_1 \lor \cdots \lor p_n \in \Theta_{\alpha}^+ \) implies \( Bp_1 \lor \cdots \lor Bp_n \in S_\delta \).
2. \( \neg p_1 \lor \cdots \lor \neg p_n \in \Theta_{\alpha}^+ \) implies \( \neg Bp_1 \lor \cdots \lor \neg Bp_n \in S_\delta \).
3. \( p_1 \land \cdots \land p_n \in \Theta_{\alpha}^+ \) implies \( \neg Bp_1 \lor \cdots \lor \neg Bp_n \in S_\delta \).
4. If \( I \) is a minimal model of \( P_{\gamma B} \cup S_\delta \), then \( I \models Cons(P_{\gamma B} \cup S_\delta, \Theta_{\alpha}) \).

The proof of this lemma needs the following two auxiliary lemmas.

**Lemma A.11**. For all \( \alpha \), if \( \neg q \in \Theta_{\alpha}^- \), then \( q \in \Theta_{\alpha}^+ \).

**Proof.** By induction on \( \alpha \).

\( \alpha = 0 \): trivial.

\( \alpha = \gamma + 1 \): Assume that \( \neg q \in \Theta_{\gamma}^- \). There are two possibilities: either \( q \in \Theta_{\gamma}^+ \) or \( q \not\in \Theta_{\gamma}^+ \). In the first case, the lemma follows immediately because \( q \in \Theta_{\gamma}^+ \subseteq \Theta_{\gamma + 1}^+ = \Theta_{\alpha}^+ \). In the second case, define \( B' = \Theta_{\gamma} \cup \{q\} \). Note that:

1. \( B' \) is a belief state.
2. \( Cons(P_{\gamma B} \cup S, B') = Cons(P_{\gamma B} \cup S, \Theta_{\gamma}) \cup Cons(S, \{q\}) = Cons(P_{\gamma B} \cup S, \Theta_{\gamma}) \cup \{\neg q\} \).
3. \( Cons(P_{\gamma B} \cup S, B') \not\models \neg q \) (from (b)).
4. \( Cons(P_{\gamma B} \cup S, B') \) must be inconsistent; otherwise, from (c) and Lemma 2.2(vi), we would have \( \neg q \not\in \Theta_{\gamma + 1}^- = \Theta_{\alpha}^- \), which contradicts the assumption.
5. \( Cons(P_{\gamma B} \cup S, \Theta_{\gamma}) \not\models q \) (from (b), (d), and the deduction theorem).
6. \( q \in \Theta(\Theta_{\gamma})^+ = \Theta_{\alpha}^+ \) (from the previous point, by Lemma 2.2(v)).

Finally, let \( \alpha \) be a limit ordinal: If \( \neg q \in \Theta_{\alpha}^- \), then there must be a \( \gamma \) such that \( \gamma < \alpha \) and \( \neg q \in \Theta_{\gamma}^- \). By induction hypothesis, derive \( q \in \Theta_{\gamma}^+ \), which implies \( q \in \Theta_{\alpha}^+ \). \( \square \)

**Lemma A.12.** Let \( I \) be a minimal model of \( P_{\gamma B} \cup S_\delta \) such that \( I \models p_1 \land \cdots \land p_n \).
where \( p_1, \ldots, p_n \in \mathcal{H} \). For all \( B \), if \( \text{Cons}(P_B \cup \text{SA}, B) \supseteq \{ \neg a \mid a \in \mathcal{H} \cup \mathcal{H}' \text{ and } I \not\models a \} \), then \( \text{Cons}(P_B \cup \text{SA}, B) \models p_1 \land \cdots \land p_n \).

**Proof.** Suppose not. Then, for some \( p_1, \ldots, p_n \in \mathcal{H} \), some \( I \) and some \( B \) we have:

(a) \( I \models p_1 \land \cdots \land p_n \).
(b) \( \text{Cons}(P_B \cup \text{SA}, B) \supseteq \{ \neg A \mid A \in \mathcal{H} \cup \mathcal{H}' \text{ and } I \not\models A \} \).
(c) \( \text{Cons}(P_B \cup \text{SA}, B) \not\models p_1 \land \cdots \land p_n \).

By (c), \( \text{Cons}(P_B \cup \text{SA}, B) \) must have a model \( I' \) that falsifies some \( p_h \) (\( h \in \{1, \ldots, n\} \)). By (b), \( I' \) is a model of \( \{ \neg A \mid A \in \mathcal{H} \cup \mathcal{H}' \text{ and } I \not\models A \} \) and hence, \( I' \subseteq I \). Thus Lemma A.9 can be applied to conclude \( (I' \cap \mathcal{H}) = (I \cap \mathcal{H}) \). However, this is absurd, because \( p_h \) must be true in \( I \) and false in \( I' \). \( \square \)

**Proof of Lemma A.10.** (By induction on \( \alpha \)). The cases where \( \alpha = 0 \) or \( \alpha \) is a limit ordinal are easy and are left to the reader. Now suppose that \( \alpha = \beta + 1 \).

(i) If \( p_1 \lor \cdots \lor p_n \in \Theta_\beta^+ \), then \( \text{Cons}(P_B \cup \text{SA}, \Theta_\beta) \models p_1 \lor \cdots \lor p_n \) (by Lemma 2.2(v)). Therefore, every minimal model of \( P_B \cup \text{SA} \)—being a model of \( \text{Cons}(P_B \cup \text{SA}, \Theta_\beta) \) by induction hypothesis (iv)—must satisfy \( p_1 \lor \cdots \lor p_n \). It follows that \( p_1 \lor \cdots \lor p_n \in \text{ECWA}(P_B \cup \text{SA}) \) and hence, \( Bp_1 \lor \cdots \lor Bp_n \in S_{\delta+1} = S_\delta \).

(ii) Similar to the proof of (i).

(iii) Suppose that (iii) does not hold, that is, \( p_1 \land \cdots \land p_n \in \Theta_\alpha^- \), but \( \neg BP_1 \lor \cdots \lor \neg BP_n \not\in S_\delta \). Then \( \neg p_1 \lor \cdots \lor \neg p_n \not\in \text{ECWA}(P_B \cup \text{SA}) \), that is, there exists a minimal model \( I \) of \( P_B \cup \text{SA} \), where \( p_1 \land \cdots \land p_n \) is true. Define

\[
B_1 = \langle \{ \neg q \mid q \in \mathcal{H} \text{ and } I \models \neg B_q \}, \{ q \mid q \in \mathcal{H} \text{ and } I \models q \} \rangle,
\]

\[
B_2 = \Theta_\beta \cup B_1.
\]

Now we prove three facts that imply \( p_1 \land \cdots \land p_n \not\in \Theta_\alpha^- \).

(a) \( B_2 \) is a belief state, that is, \( B_2^+ \cap B_2^- = \emptyset \). In order to prove this, it should be shown that:

1. \( \Theta_\beta^+ \cap \Theta_\beta^- = \emptyset \).
2. \( B_2^+ \cap B_2^- = \emptyset \).
3. \( \Theta_\beta^+ \cap B_2^- = \emptyset \).
4. \( \Theta_\beta^- \cap B_2^+ = \emptyset \).

(1) and (2) are left to the reader. (3) If \( q \in \Theta_\beta^+ \), then \( Bq \in S_\delta \) (by induction hypothesis (i)) and hence, \( q \) must be true in every minimal model of \( P_B \cup S_\delta \), including \( I \). It follows that \( q \) cannot belong to \( B_2^- \). (4) If \( \neg q \in \Theta_\beta^- \), then \( q \in \Theta_\beta^+ \), by Lemma A.11. By induction hypothesis (i), \( Bq \in S_\delta \) and hence, \( I \models Bq \). Consequently, \( \neg q \not\in B_2^+ \).

(b) \( \text{Cons}(P_B \cup \text{SA}, B_2) \) is consistent. This fact is proved by showing that \( I \) is a model of \( \text{Cons}(P_B \cup \text{SA}, B_2) \). In fact, \( \text{Cons}(P_B \cup \text{SA}, B_2) \) can be rewritten as \( \text{Cons}(P_B \cup \text{SA}, \Theta_\beta) \cup \text{Cons}(\text{SA}, B_1) \). By induction hypothesis (iv), we have \( I \models \text{Cons}(P_B \cup \text{SA}, \Theta_\beta) \). We are left to show that \( I \models \text{Cons}(\text{SA}, B_1) \), which follows by noting that \( \text{Cons}(\text{SA}, B_1) \) is just the set of negative literals satisfied by \( I \), that is, \( \text{Cons}(\text{SA}, B_1) = \{ \neg a \mid a \in \mathcal{H} \cup \mathcal{H}' \text{ and } I \models \neg a \} \).
(c) \(\text{Cons}(P_B \cup SA, B_2) \vdash p_1 \land \cdots \land p_n\). It follows from Lemma A.12 by noting that
\[
\text{Cons}(P_B \cup SA, B_2) \supseteq \text{Cons}(SA, B_1) = \{\neg a \mid a \in H \cup H' \text{ and } I \models \neg a\}.
\]
The previous facts, by Lemma 2.2(vi), imply \(p_1 \land \cdots \land p_n \notin (\Theta_\alpha)^-\). A contradiction.

(iv) Let \(I\) be a minimal model of \(P_B \cup S_5\) and let \(\phi \in \text{Cons}(P_B \cup SA, \Theta_\alpha)\).
There are four possibilities:

(a) \(\phi \in P_B\). Then obviously \(I \models \phi\).

(b) \(\phi = \neg q_1 \lor \cdots \lor \neg q_m\). Then \(\phi \in \text{Cons}(P_B \cup SA, \Theta_\alpha)\) only if \(q_1 \land \cdots \land q_m \in \Theta_\alpha^-\) which implies \(\neg Bq_1 \lor \cdots \lor \neg Bq_m \in S_5\) (by (iii)). It follows that \(\neg q_1 \lor \cdots \lor \neg q_m \in \text{ECWA}(P_B \cup S_5)\) and hence, \(\phi\) is true in all the minimal models of \(P_B \cup S_5\), including \(I\).

(c) \(\phi = Bq_1 \lor \cdots \lor Bq_m\). Then \(\phi \in \text{Cons}(P_B \cup SA, \Theta_\alpha)\) only if \(q_1 \lor \cdots \lor q_m \in \Theta_\alpha^+\) which implies \(Bq_1 \lor \cdots \lor Bq_m \in S_5\) (by (i)). It follows that \(I\), being a model of \(S_5\), must satisfy \(\phi\).

(d) \(\phi = \neg Bq_1 \lor \cdots \lor \neg Bq_m\). The proof that \(I\) satisfies \(\phi\) is similar to the previous case.

Lemma A.13 (Completeness). Let \(B\) be the least fixed point of \(\Theta\). For all ordinals \(\alpha\):

(i) \(Bp_1 \lor \cdots \lor Bp_n \in S_\alpha\) implies \(p_1 \lor \cdots \lor p_n \in B^+\).

(ii) \(\neg Bp_1 \lor \cdots \lor \neg Bp_n \in S_\alpha\) implies \(p_1 \land \cdots \land p_n \in B^-\).

(iii) \(S_\alpha \subseteq \text{Cons}(P_B \cup SA, B)\).

The proof of Lemma A.13 requires the following Lemma:

Lemma A.14. Assume that \(\text{Cons}(P_B \cup SA) \supseteq S_\beta\) holds, for some \(B\) and \(\beta\). If \(I\) is a minimal model of \(\text{Cons}(P_B \cup SA, B)\) and \(I \models p_1 \land \cdots \land p_n\), then \(P_B \cup S_\beta\) has a minimal model \(I'\) such that \(I' \models p_1 \land \cdots \land p_n\).

PROOF. Assume that \(I\) is a minimal model of \(\text{Cons}(P_B \cup SA, B)\) and that \(I \models p_1 \land \cdots \land p_n\). Let \(NC\) be the set of negative clauses of \(\text{Cons}(SA, B)\) and \(PC\) be the set of positive clauses of \(\text{Cons}(SA, B)\). Note that

\[
\text{Cons}(P_B \cup SA, B) = \text{Cons}(P_B \cup SA, B) \cup S_\beta \\
= P_B \cup \text{Cons}(SA, B) \cup S_\beta \\
= P_B \cup PC \cup NC \cup S_\beta.
\]

It is easy to see that every minimal model of \(P_B \cup PC \cup NC \cup S_\beta\) is also a minimal model of \(P_B \cup PC \cup S_\beta\), because \(NC\) contains only negative clauses. Thus, \(I\) is a minimal model of \(P_B \cup PC \cup S_\beta\). Note that \(PC \cup S_\beta\) is a \(P\)-state. \(I\) is also a model of \(P_B \cup S_\beta\) and hence, there must be a minimal model \(I'\) of \(P_B \cup S_\beta\) such
that $I' \subseteq I$. By Lemma A.9, we have $I' \cap \mathcal{H} = I \cap \mathcal{H}$ and hence, the assumption $I \models p_1 \land \cdots \land p_n$ implies $I' \models p_1 \land \cdots \land p_n$. □

**Proof of Lemma A.13** (By induction on $\alpha$). The base case and the limit case are trivial. We prove only the step case. Let $\alpha = \beta + 1$.

(i) Assume $Bp_1 \lor \cdots \lor Bp_n \in S_\alpha$. We show that this implies $p_1 \lor \cdots \lor p_n \in B^+$. By definition of $S_{\beta+1}, p_1 \lor \cdots \lor p_n \in ECWA(P_\beta \cup S_\beta)$ and hence,

$$P_\beta \cup S_\beta \models p_1 \lor \cdots \lor p_n \quad (1)$$

Note that $Cons(P_\beta \cup SA, B) \supseteq P_\beta$ and (by induction hypothesis (iii)) $Cons(P_\beta \cup SA, B) \supseteq S_\beta$. It follows that $Cons(P_\beta \cup SA, B) \supseteq P_\beta \cup S_\beta$; hence, by (1), $Cons(P_\beta \cup SA, B) \models p_1 \lor \cdots \lor p_n$. By Lemma 2.2 (vii) this implies $p_1 \lor \cdots \lor p_n \in B^+$.

(ii) Suppose that (ii) does not hold, that is, there exists a clause $\neg Bp_1 \lor \cdots \lor \neg Bp_n$ in $S_\alpha$ such that $p_1 \land \cdots \land p_n \notin B^-$. By Lemma 2.2(viii), there must be a belief state $B'$ such that:

(a) $B' \supseteq B$.
(b) $Cons(P_\beta \cup SA, B') \models p_1 \land \cdots \land p_n$.
(c) $Cons(P_\beta \cup SA, B')$ is consistent.

Let $I$ be a minimal model of $Cons(P_\beta \cup SA, B')$. The following facts hold:

(d) $I \models p_1 \land \cdots \land p_n$ (from (b)).
(e) $Cons(P_\beta \cup SA, B') \supseteq Cons(P_\beta \cup SA, B) \supseteq S_\beta$ (by Lemma 2.2(i) and induction hypothesis (iii)).
(f) There exists a minimal model $I'$ of $P_\beta \cup S_\beta$ that satisfies $p_1 \land \cdots \land p_n$. It follows from (e), (d), and Lemma A.14

(g) $\neg p_1 \lor \cdots \lor \neg p_n \notin ECWA(P_\beta \cup S_\beta)$ (by (f)).

However, then $\neg Bp_1 \lor \cdots \lor \neg Bp_n$ should not belong to $S_{\beta+1} = S_\alpha$. A contradiction.

(iii) Let $\varphi$ be any member of $S_\alpha$. We will show that $\varphi \in Cons(P_\beta \cup SA, B)$. There are two possibilities:

(a) $\varphi = Bp_1 \lor \cdots \lor Bp_n$. In this case, by (i), $p_1 \lor \cdots \lor p_n \in B^+$. It follows, from the definition of SA, that $Bp_1 \lor \cdots \lor Bp_n \in Cons(P_\beta \cup SA, B)$, that is, $\varphi \in Cons(P_\beta \cup SA, B)$.

(b) $\varphi = \neg Bp_1 \lor \cdots \lor \neg Bp_n$. In this case, by (ii), $p_1 \land \cdots \land p_n \in B^-$ and hence, $\neg p_1 \lor \cdots \lor \neg p_n \in Cons(P_\beta \cup SA, B)$. By Lemma 2.2(vii), this implies $\neg p_1 \lor \cdots \lor \neg p_n \in B^+$. From the definition of SA, it follows that $\neg Bp_1 \lor \cdots \lor \neg Bp_n \in Cons(P_\beta \cup SA, B)$, that is, $\varphi \in Cons(P_\beta \cup SA, B)$.

This is true of arbitrary members of $S_\alpha$, so we derive $S_\alpha \subseteq Cons(P_\beta \cup SA, B)$. □

**Theorem 5.1.** Let $P$ be a disjunctive program and let $B = \text{lfp}(\Theta_{P_\beta \cup SA})$.

(i) $B$ is the least GSE of $P_\beta \cup SA$.
(ii) $STN(P) = B \cap \langle DHB, CHB \rangle$. 


Proof. (i) \( P \cup S \) is consistent. Then \( P \cup S \) must have a minimal model \( I \).

We have:

(a) for all \( \alpha \), \( I \models Cons(P \cup SA, \Theta_\alpha) \) (by Lemma A.10(iv)).
(b) \( I \models Cons(P \cup SA, B) \) (because \( B = \Theta_\alpha \), for some \( \alpha \)).
(c) \( (I, B) \models P \cup SA \) (by Lemma 2.2(ii)).
(d) \( P \cup SA \) is \( B \)-consistent.
(e) \( B \) is a GSE of \( P \cup SA \) (from (d) and Lemma A.2(viii)).
(f) \( B \) is the least GSE of \( P \cup SA \) (from (e) and Theorem 2.4).

(ii) It follows easily from Lemma A.10(i, iii) and Lemma A.13(i, ii).

Ignorance Tests

We assume the reader to be familiar with the definitions and results of [35].

Lemma A.15. Let \( p, q \) be distinct atoms, \( P \) be a program, and \( d \) be a disjunction in DHB. If \( p \lor q \in P \), then, under the strong semantics:

(i) \( Q \) is a basis for \( d \) in \( P \) only if \( Q \) is a basis for \( d \) in \( P \cup \{ p \} \).

(ii) \( Q \) is a basis for \( d \) in \( P \cup \{ p \} \) only if \( d \) has a basis \( Q' \) in \( P \) such that \( Q' = Q \lor \{ q \} \).

Under the weak semantics:

(iii) \( Q \) is a basis (resp. a weak basis) for \( d \) in \( P \) only if \( Q \) is a basis (resp. a weak basis) for \( d \) in \( P \cup \{ p \} \).

(iv) \( Q \) is a basis for \( d \) in \( P \cup \{ p \} \) only if \( Q \) is a (possibly weak) basis for \( d \) in \( P \).

(v) \( Q \) is a weak basis for \( d \) in \( P \cup \{ p \} \) only if \( Q \) is a weak basis for \( d \) in \( P \).

Proof (Sketch). (i) and (iii) are trivial, since every derivation from \( P \) can be done also in \( P \cup \{ p \} \).

(ii) Let \( Q \) be a basis for \( d \) in \( P \cup \{ p \} \). Then there exists a weak derivation from \( P \cup \{ p \} \), having the form \( Q_1, \ldots, Q_n \), where \( Q_1 = d \) and \( Q_n = Q \). Obtain a similar derivation \( Q'_1, \ldots, Q'_n \), from \( P \), by the following method:

- Set \( Q'_1 = d \).
- If \( Q_{i+1} \) is obtained from \( Q_i \) by means of a program rule \( R \neq p \) and a derivation rule \( S_j \), then construct \( Q'_{i+1} \) from \( Q'_i \) by means of \( R \) and \( S_j \).
- If \( Q_{i+1} \) is obtained from \( Q_i \) by means of the program rule \( p \) (and hence \( S_1 \) is the only applicable inference), then construct \( Q'_{i+1} \) from \( Q'_i \) by means of \( S_2 \) and \( p \lor q \).

In the last case, for some \( d \) and \( d' \), we have

\[
Q_{i+1} = Q_i \setminus \{d\},
Q'_{i+1} = (Q'_i \setminus \{d\}) \cup \{-q\}.
\]

From this fact, by a simple induction argument, it can be proved that either \( Q'_i = Q_i \) or \( Q'_i = Q_i \cup \{-q\} \), for \( i = 1, \ldots, n \). Let \( Q' = Q'_n \). For what we said before, we have that either \( Q' = Q_n \lor q \) or \( Q' = Q_n \lor \{q\} \). This completes the proof of (ii).
(iv) and (v) Let Q be a possibly weak basis for \( d \) in \( P \cup \{ p \} \). Then there exists a weak derivation from \( P \cup \{ p \} \), having the form \( Q_1, \ldots, Q_n \), where \( Q_1 = d \) and \( Q_n = Q \). Obtain a similar derivation \( Q'_1, \ldots, Q'_n \), from \( P \), by the following method:

- Set \( Q'_1 = d \)
- If \( Q_{i+1} \) is obtained from \( Q_i \) by means of a program rule \( R \neq p \) and a derivation rule \( W_j \), then construct \( Q'_{i+1} \) from \( Q'_i \) by means of \( R \) and \( W_j \).
- If \( Q_{i+1} \) is obtained from \( Q_i \) by means of the program rule \( p \) (and hence \( W_1 \) is the only applicable inference), then construct \( Q'_{i+1} \) from \( Q'_i \) by means of \( W_2 \) and \( p \lor q \).

By a simple induction argument, it can be proved that \( Q_i = Q'_i \), for \( i = 1, \ldots, n \). It follows that \( Q'_n = Q_n = Q \) and hence, \( Q \) is a basis for \( d \) in \( P \) (or a weak basis, if \( W_2 \) has been applied).

Lemma 5.2. Let \( SEM \) be the weak well-founded semantics for disjunctive programs and let \( p, q \) be distinct atoms. Then, for all \( P \),

\[
SEM(P \cup \{ p \lor q \}) \subseteq SEM(P \cup \{ p \lor q \} \cup \{ p \}).
\]

PROOF. Since every disjunction \( d \) in \( SEM(P \cup \{ p \lor c \})^+ \cup SEM(P \cup \{ p \lor c \})^- \) has a level \( l(d) \), it suffices to show that, for all ordinals \( l \),

\[
SEM(P \cup \{ p \lor q \})_l \subseteq SEM(P \cup \{ p \lor q \} \cup \{ p \}),
\]

where \( B |_l \) is an abbreviation for \( \langle B^+ \cap \{ d \mid l(d) = l \}, B^- \cap \{ d \mid l(d) = l \} \rangle \). The proof is by induction on \( l \).

\((l = 0)\): Trivial (note that \( SEM(P \cup \{ p \lor c \})_0 = \emptyset \)).

\((l > 0)\): We have to show that:

(a) \( SEM(P \cup \{ p \lor q \})_l^+ \subseteq SEM(P \cup \{ p \lor q \} \cup \{ p \})^+ \).

(b) \( SEM(P \cup \{ p \lor q \})_l^- \subseteq SEM(P \cup \{ p \lor q \} \cup \{ p \})^- \).

(a) If \( d \in SEM(P \cup \{ p \lor c \})_l^+ \), then \( d \) has a false child in the associated global tree, such that \( l(d') < l(d) = l \). By Lemma A.15(iii), \( d' \) is a child of \( d \) also in the global tree associated to \( P \cup \{ p \lor c \} \cup \{ p \} \). Moreover, by induction hypothesis, \( d' \) is false in \( SEM(P \cup \{ p \lor q \} \cup \{ p \}) \). It follows that \( d \) is true in \( SEM(P \cup \{ p \lor q \} \cup \{ p \}) \), i.e., \( d \in SEM(P \cup \{ p \lor q \} \cup \{ p \})^+ \). This proves (a).

(b) We show that the negation of (b) leads to a contradiction. Suppose that, for some \( d, d \in SEM(P \cup \{ p \lor c \})_l^- \), but \( d \notin SEM(P \cup \{ p \lor q \} \cup \{ p \})^- \). Then, in the global tree associated to \( P \cup \{ p \lor q \} \cup \{ p \} \), \( d \) must have a child \( d' \), which is not true, and whose level is strictly smaller than \( l \). By Lemma A.15(iv, v), \( d' \) is a child of \( d \) also in the global tree associated to \( P \cup \{ p \lor q \} \). Moreover, by the contrapositive of the induction hypothesis, \( d' \) is not true in \( SEM(P \cup \{ p \lor c \}) \). It follows that \( d \) is not false in \( SEM(P \cup \{ p \lor c \}) \). A contradiction. □

Lemma 5.3. Let \( SEM \) be the strong well-founded Semantics, let \( P \) be a disjunctive program, and let \( p, q, c \) be distinct atoms, where \( c \) does not occur in the heads of the rules of \( P \). Finally, let \( P' = P \cup \{ p \lor q, p \lor c \} \). Then some of the following conditions hold:
The proof of Lemma 5.3 needs the following result:

**Lemma A.16.** Under the assumptions of Lemma 5.3, we have that, for all active strong derivations \( d \lor c, Q_1, \ldots, Q_n \) from \( P' \), where \( Q_n \) does not contain \( \neg p \), there exists an active strong derivation \( d, Q'_1, \ldots, Q'_m \) from \( P' \cup \{p\} \) such that \( Q'_m = Q_n \).

**Proof (Sketch).** By induction on \( n \).

\((n = 0)\): The lemma is vacuously true: since \( d \lor c \) is not empty, all of its active derivations have nonzero length.

\((n > 0)\): There are three possibilities:

- \( Q_1 \) is obtained by rewriting \( d \lor c \) with \( p \lor c \) and \( S_1 \). Then \( p \) must be contained in \( d \) and hence, we can define \( Q'_1 \) by rewriting \( d \) with \( p \) and \( S_1 \). Note that \( Q'_1 = Q_1 = \square \), so the lemma holds in this case.
- \( Q_1 \) is obtained by rewriting \( d \lor c \) with \( p \lor c \) and \( S_2 \). In this case there are two possibilities, corresponding to the possible applications of \( p \lor c \) and \( S_2 \): either \( Q_1 = \{\neg p\} \) or \( Q_1 = \{\neg c\} \). The first case contradicts the hypothesis, so we have \( Q_1 = \{\neg c\} \). This derivation is possible only if \( p \) is contained in \( d \lor c \). Since \( p \neq c \), it follows that \( p \) is contained in \( d \). Thus, we can rewrite \( d \) with \( p \lor c \) and \( S_2 \), obtaining \( Q'_1 = \{\neg c\} = Q_1 \).
- \( Q_1 \) is obtained by rewriting \( d \lor c \) with a program rule \( R \neq p \lor c \) and a derivation rule \( S_j \). Also \( d \) can be rewritten by means of \( R \) and \( S_j \), because, by hypothesis, \( c \) does not occur in the head of \( R \). Let \( Q'_1 \) be the resulting goal. Note that for each extended literal \( l' \in Q'_1 \), there is a corresponding literal \( l \in Q_1 \), such that either \( l \vdash l' \) or \( l = l' \lor c \). From \( Q_1, \ldots, Q_n \), we can extract a strong active derivation starting from \( l \). Let \( Q(l) \) be the corresponding basis. By induction hypothesis, \( l' \) has an active derivation with basis \( Q(l') = Q(l) \). Construct \( Q'_2, \ldots, Q'_m \) by merging these derivations, for all \( l' \in Q'_1 \). It is easy to see that \( Q'_m = \bigcup_{l' \in Q'_1} Q(l') = \bigcup_{l \in Q_1} Q(l) = Q_n \). This completes the proof. \( \square \)

**Proof of Lemma 5.3.** There are two possibilities:

\((a)\) \( p \in \text{SEM}(P') \) (i.e., \( p \) is false in \( \text{SEM}(P') \)).

\((b)\) \( p \not\in \text{SEM}(P') \) (i.e., \( p \) is not false in \( \text{SEM}(P') \)).

\((a)\) In this case \( q \) is true in \( \text{SEM}(P') \). In fact, \( q \) has a basis \( \{\neg p\} \) (obtained through \( p \lor q \) and \( S_2 \)), so \( p \) is a child of \( q \) in the global tree. Then \( q \) has a false child and hence, \( q \) is true. It follows easily that

\[ \text{SEM}(P') = \text{SEM}(P' \cup \{q\}) \].

\((b)\) In this case, we will show that

\[ \text{SEM}(P') \subseteq \text{SEM}(P' \cup \{p\}) \].
It suffices to show that, for all ordinals $l$,

$$SEM(P')_l \subseteq SEM(P' \cup \{p\})$$

(see Lemma 5.2). The proof is by induction on $l$.

($l = 0$): Trivial (note that $SEM(P')_0 = \emptyset$).

($l > 0$): Let $d$ be any member of $SEM(P')_l^+$. Since $d$ is true, it must have a false child $d'$, in the associated global tree, such that $l(d') < l(d) = l$. By lemma A.15(i), $d'$ is a child of $d$ in the global tree associated to $P' \cup \{p\}$. Moreover, by induction hypothesis, $d'$ is false in $SEM(P' \cup \{p\})$. It follows that $d$ is true in $SEM(P' \cup \{p\})$, i.e., $d \in SEM(P' \cup \{p\})^+$. This proves that $SEM(P')_l^+ \subseteq SEM(P' \cup \{p\})^+$.

Now assume that, for some $d, d' \in SEM(P')_l^-$, but $d \notin SEM(P' \cup \{p\})^-$. We show that this leads to a contradiction.

Since $d$ is not false in $SEM(P' \cup \{p\})$, $d$ must have a child $d_1$ in the associated global tree, such that $d_1$ is not true.

By Lemma A.15(ii), $d$ has a corresponding child $d_2$ in the global tree of $P'$, such that either $d_2 = d_1$ or $d_2 = d_1 \lor c$.

We have $d_2 \neq d_1$. In fact, by assumption, $d$ is false in $SEM(P')$ and hence, $d_2$ must be true in $SEM(P')$. By induction hypothesis, this implies that $d_2$ is true in $SEM(P' \cup \{p\})$. Since $d_1$ is not true in $SEM(P' \cup \{p\})$, we have $d_2 \neq d_1$.

It follows that $d_2 = d_1 \lor c$. Since $d_2$ is true in $SEM(P')$, $d_2$ must have a false child $d_3$ in the corresponding global tree. Now, there are two possibilities: either $d_3$ contains $p$ or $d_3$ does not contain $p$.

The first case is impossible: since $d_3$ is false, all of its disjuncts are false, too, while $p$ is not false by (b).

Conclude that $d_3$ does not contain $p$. Then, by Lemma A.16, we have that $d_3$ is a child of $d_1$ also in the global tree associated to $P' \cup \{p\}$. Moreover, by induction hypothesis, $d_3$ is false in $SEM(P' \cup \{p\})$. It follows that $d_1$ must be true in $SEM(P' \cup \{p\})$. A contradiction. □

Theorem 5.4. If $p, q$ are independent in $P_{\varphi B}$, then

$$STN(P) \subseteq STN(P \cup \{p\}).$$

The proof of this theorem needs the following two technical lemmas.

Lemma A.17. Let $S, S'$ be $P$-states such that $S \subseteq S'$. Then

$$Ord(ECWA(P_{\varphi B} \cup S)) \subseteq Ord(ECWA(P_{\varphi B} \cup S')).$$

Proof. Suppose not, that is, suppose that there exists an ordinary sentence $\varphi \in ECWA(P_{\varphi B} \cup S)$ that is false in some minimal model $I'$ of $P_{\varphi B} \cup S'$. $I'$ is also a model of $P_{\varphi B} \cup S$ and hence, there exists a minimal model $I$ of $P_{\varphi B} \cup S$ such that $I \subseteq I'$. By lemma A.9 we have $(I \cap \mathcal{H}) = (I' \cap \mathcal{H})$. Then $\varphi$ must be given the same truth value by $I$ and $I'$, that is, $\varphi$ must be false in $I$. However, this implies, $\varphi \notin ECWA(P_{\varphi B} \cup S)$, contradicting the assumption. □

Lemma A.18. If $p, q$ are independent in $P_{\varphi B}$, then for all $P$-states $S$,

$$Ord(ECWA(P_{\varphi B} \cup S)) \subseteq Ord(ECWA(P_{\varphi B} \cup \{p\} \cup S)).$$
PROOF. Suppose not, that is, suppose that there exists an ordinary sentence $\varphi \in ECWA(P_{cB} \cup S)$ that is false in some minimal model $I$ of $P_{cB} \cup \{p\} \cup S$. Since $p, q$ are independent in $P_{cB}, I$ must be a minimal model of $P_{cB} \cup S$. It follows that $\varphi \notin ECWA(P_{cB} \cup \{p \lor q\} \cup S)$, which is absurd. □

PROOF OF THEOREM 5.4. Let $S_0, S_1, \ldots, S_{\alpha}, \ldots$ be the transfinite sequence converging to the stationary state of $P$ and let $S'_0, S'_1, \ldots, S'_{\alpha}, \ldots$ be the corresponding sequence converging to the stationary state of $P \cup \{p\}$. It suffices to show that for all $\alpha$, $S_{\alpha} \subseteq S'_{\alpha}$. The proof is by induction on $\alpha$. The base case and the limit case are trivial. We prove only the step case. Assume $\alpha = \beta + 1$. We have

$$S_{\alpha} = \{\varphi \mid \varphi \in ECWA(P_{cB} \cup S_{\beta})$$
and $\varphi$ has the form: $B_{p_1} \lor \cdots \lor B_{p_n}$
or $\neg B_{p_1} \lor \cdots \lor \neg B_{p_n}\}
\subseteq \{\varphi \mid \varphi \in ECWA(P_{cB} \cup S'_{\beta})$
and $\varphi$ has the form: $B_{p_1} \lor \cdots \lor B_{p_n}$
or $\neg B_{p_1} \lor \cdots \lor \neg B_{p_n}\}$ (by Lemma A.17 since, by induction hypothesis, $S_{\beta} \subseteq S'_{\beta}$)
\subseteq \{\varphi \mid \varphi \in ECWA([P \cup \{p\}]_{cB} \cup S'_{\beta})$
and $\varphi$ has the form: $B_{p_1} \lor \cdots \lor B_{p_n}$
or $\neg B_{p_1} \lor \cdots \lor \neg B_{p_n}\}$ (by Lemma A.18)
= $S'_{\beta + 1}$
= $S'_{\alpha}$.
□

Lemma 5.4. Let $p, q, c, c'$ be distinct atoms of $\mathcal{H}$, with the exception of $c$ and $c'$, which may be equal. Let $P$ be a disjunctive program, whose rule contain neither $c$ nor $c'$ in their heads. Then $p, q$ are independent in

$$P' = [P \cup \{(p \lor q), (p \lor c), (q \lor c')\}]_{cB}.$$

PROOF. We have to show that, for all $P$-states $S$:

(a) $MM(P' \cup S) \subseteq MM(P' \cup \{p\} \cup S) \cup MM(P' \cup \{q\} \cup S)$.
(b) $MM(P' \cup S) \supseteq MM(P' \cup \{p\} \cup S) \cup MM(P' \cup \{q\} \cup S)$.

(a) Let $I$ be an arbitrary member of $MM(P' \cup S)$. It suffices to show that $I \in MM(P' \cup \{p\} \cup S) \cup MM(P' \cup \{q\} \cup S)$. By assumption, $I$ satisfies either $p$ or $q$. Assume $I \models p$, without loss of generality. Then $I$ is a model of $P' \cup \{p\} \cup S$. All the models of $P' \cup \{p\} \cup S$ are also models of $P' \cup S$ and hence, none of them can be strictly smaller than $I$, by assumption. It follows that $I \in MM(P' \cup \{p\} \cup S)$ and hence, $I \in MM(P' \cup \{p\} \cup S) \cup MM(P' \cup \{q\} \cup S)$.
(b) Let $I$ be an arbitrary member of $MM(P' \cup \{p\} \cup S) \cup MM(P' \cup \{q\} \cup S)$. It suffices to show that assuming $I \notin MM(P' \cup S)$ leads to a contradiction. There are two possibilities:

(b.1) $I \in MM(P' \cup \{p\} \cup S)$.  
(b.2) $I \in MM(P' \cup \{q\} \cup S)$.

First assume (b.1). If $I \notin MM(P' \cup S)$, then there exists an interpretation $I' \in MM(P' \cup S)$ such that $I' \cup I$. $I'$ cannot be a model of $P'$; otherwise (b.1) would not be satisfied. Therefore, in order to satisfy $p \lor q$, $I'$ must satisfy $q$, i.e., $q \in I'$, which implies $q \in I$. Similarly, in order to satisfy $p \lor c$, $I'$ must satisfy $c$ and hence, $c \in I$. Now consider the interpretation $I \setminus \{c\}$, which is strictly smaller than $I$. We have:

(c) $I \setminus \{c\} \models P_{\lor B}$. In fact, by assumption, $c$ does not occur in the heads of the rules of $P_{\lor B}$. Consequently, $I \setminus \{c\}$ and $I$ satisfy the same heads. Similarly, $\neg c$ does not occur in the bodies of the rules of $P_{\lor B}$ (because, by assumption, $c \notin \mathcal{H}$ hence $c \notin \mathcal{H}'$); therefore, $I \setminus \{c\}$ satisfies at most the same bodies that are satisfied by $I$. From the previous facts, it follows easily that $I$ satisfies a rule $R \in P_{\lor B}$ only if $I \setminus \{c\}$ satisfies $R$, too. Since $I \models P_{\lor B}$, this implies $I \setminus \{c\} \models P_{\lor B}$.  

(d) $I \setminus \{c\} \models \{(p \lor q), (p \lor c), (q \lor c')\}$. By (b.1), $I$ satisfies $p$ and hence, $p \in I$. Moreover, we have proved that $q \in I$. Since $p, q, c$ are distinct atoms (by hypothesis), we have that $I \setminus \{c\}$ contains $p$ and $q$. It follows that $I \setminus \{c\} \models \{(p \lor q), (p \lor c), (q \lor c')\}$.

(e) $I \setminus \{c\} \models P'$. Note that $P' = [P \cup \{(p \lor q), (p \lor c), (q \lor c')\}]_{\lor B} = P_{\lor B} \cup \{(p \lor q), (p \lor c), (q \lor c')\}$. Then (e) follows immediately from (c) and (d).

(f) $I \setminus \{c\} \models \{p\} \cup S$. From the hypothesis, derive $c \neq p$ and $c \notin \mathcal{H}'$. It follows that $c$ does not occur in $\{p\} \cup S$ and hence, eliminating $c$ from $I$ does not change the valuation of $\{p\} \cup S$.

By (c) and (f), we have that $I \setminus \{c\}$ is a model of $P' \cup \{p\} \cup S$. However, then, $I$ cannot be a minimal model of $P' \cup \{p\} \cup S$, which is absurd.

The proof for case (b.2) is symmetrical to the one for (b.1). 

**APPENDIX B: YET ANOTHER WELL-FOUNDED SEMANTICS**

Here we prove that the formulation of the well-founded semantics adopted in this paper is equivalent to the other formulations. First we recall some results, due to Baral and Subrahmanian:

**Proposition B.1 ([4]).**

(i) $WF(P) = \langle lfp(\Pi_P^P), -gfp(\Pi_P^P) \rangle$.

(ii) $lfp(\Pi_P^P) = \Pi_P(gfp(\Pi_P^P))$.

(iii) $gfp(\Pi_P^P) = \Pi_P(lfp(\Pi_P^P))$.

**Theorem B.1.** $lfp(\Psi_P) = WF(P)$.  

**Proof.** Let $B = lfp(\Psi_P)$. We have to show the following facts:

- $B \models P_{\lor B}$.
- $B \models P'_{\lor B}$.
- For all $c \in \mathcal{H}'$, $B \not\models \{p\} \cup S$. 
- For all $c \in \mathcal{H}'$, $B \not\models \{p\} \cup S$.
(a) \( B \subseteq WF(P) \).
(b) \( B \supseteq WF(P) \).

In order to prove (a), it suffices to show that \( WF(P) \) is a fixed point of \( \Psi_P \). We have

\[
\Psi_P(WF(P)) = \langle \Pi_P(-WF(P)^-), -\Pi_P(WF(P)^+) \rangle \quad \text{by def. of } \Psi_P \\
= \langle \Pi_P(gfp(\Pi^2_P)), -\Pi_P(lfp(\Pi^2_P)) \rangle \quad \text{by Proposition B.1(i)} \\
= \langle lfp(\Pi^2_P), -gfp(\Pi^2_P) \rangle \quad \text{by Proposition B.1(ii, iii)} \\
= WF(P).
\]

We are left to prove (b). By Proposition B.1(i), it suffices to show that:

(c) \( B^+ \supseteq lfp(\Pi^2_P) \).
(d) \( B^- \supseteq -gfp(\Pi^2_P) \).

(c) can be proved by showing that \( B^+ \) is a fixed point of \( \Pi^2_P \). We have

\[
\begin{align*}
B^+ &= \Psi_P(B)^+ \\
&= \Pi_P(-B^-) \\
&= \Pi_P(-\Psi_P(B)^-) \\
&= \Pi_P(\Pi_P(B^+)) \\
&= \Pi^2_P(B^+).
\end{align*}
\]

Similarly, (d) can be proved by showing that \( -B^- \) is a fixed point of \( \Pi^2_P \). We have

\[
\begin{align*}
-B^- &= -\Psi_P(B)^- \\
&= \Pi_P(B^+) \\
&= \Pi_P(\Psi_P(B)^+) \\
&= \Pi_P(\Pi_P(-B^-)) \\
&= \Pi^2_P(-B^-).
\end{align*}
\]

This completes the proof. \( \square \)

I am in debt to my advisor, Giorgio Levi, and to Luigia Carlucci Aiello and Michael Gelfond. I am grateful to V. S. Subrahmanian for his precious hints and comments and for suggesting the doctor's example. Many thanks to Vladimir Lifschitz for his encouragement and for pointing out the relations between \( \Psi_P \) and three-valued stable models. Special thanks to Alex Simpson and Achille Varzi, who helped me during the early development of three-valued autoepistemic logics. The suggestions of three anonymous referees helped to improve the quality of the paper.

REFERENCES


