A Model-Theoretic Reconstruction of the Operational Semantics of Logic Programs*

MORENO FALASCHI

Dipartimento di Ingegneria Elettronica e Informatica, Università di Padova,
Via Gradenigo 6/A, 35131 Padova, Italy

GIORGIO LEVI

Dipartimento di Informatica, Università di Pisa,
Corso Italia 40, 56125 Pisa, Italy

AND

MAURIZIO MARTELLI AND CATUSCIA PALAMIDESSI

Dipartimento di Informatica e Scienze dell'Informazione, Università di Genova,
Viale Benedetto XV 3, 16132 Genova, Italy

In this paper we define a new notion of truth on Herbrand interpretations extended with variables which allows us to capture, by means of suitable models, various observable properties, such as the ground success set, the set of atomic consequences, and the computed answer substitutions. The notion of truth extends the classical one to account for non-ground formulas in the interpretations. The various operational semantics are all models. An ordering on interpretations is defined to overcome the problem that the intersection of a set of models is not necessarily a model. The set of interpretations with this partial order is shown to be a complete lattice, and the greatest lower bound of any set of models is shown to be a model. Thus there exists a least model, which is the least Herbrand model and therefore the ground success set semantics. Richer operational semantics are non-least models, which can, however, be effectively defined by fixpoint constructions. The model corresponding to the computed answer substitutions operational semantics is the most primitive one (the others can easily be obtained from it). 4 1993 Academic Press, Inc.

1. INTRODUCTION

The least Herbrand model semantics was originally proposed (van Emde Emden and Kowalski, 1976) as the correct declarative semantics for

* Partially supported by the ESPRIT Basic Research Action P3020 ("Integration").
definite horn clause logic (HCL) programs. This characterization is meaningful from a purely logical point of view and has some real minimality properties, as we will point out in the following.

Unfortunately, if we look at the problem from a programming language point of view, this kind of semantics is not rich enough to model important properties of HCL programs.

In general the declarative semantics of a programming language should be equivalent to the operational semantics, which, in turn, is strongly influenced by what are called observable properties. A specific operational behaviour (captured by the operational semantics) can be relevant or not depending on which kind of properties we are interested in. These same properties should have a counterpart in the declarative semantics and this may require defining a richer model-theoretic semantics. Of course, this semantics should also encompass the classical approach, i.e., the least Herbrand model semantics. A first partial solution to this problem was given by Clark (1979). We extended this solution in Falaschi et al. (1988, 1989) by defining two semantics modeling important observable properties and by characterizing them as fixpoints. The same motivation can be found in Gaifman and Shapiro (1989a, b), which introduce a proof-theoretic approach, able to deal with compositionality.

In this paper we outline a formal framework to describe and compare different semantics. Our approach is essentially model-theoretic rather than proof-theoretic. In the resulting construction different semantics correspond to different models, including the standard semantics which is still the least model.

Section 2 defines various operational semantics with different (already proposed) notions of observable properties, such as the ground success set, the set of atomic consequences, and the computed answer substitutions. It also discusses the induced program equivalence relations.

Section 3 describes the new notions of interpretations and models. Interpretations contain non-ground atoms and include standard Herbrand interpretations. The notion of truth extends the classical one to account for non-ground formulas in the interpretations.

The various operational semantics are all models. However, one relevant property of Herbrand models does not hold: the intersection of a set of models is not necessarily a model.

In order to overcome this problem a new partial order on interpretations is defined (Section 4). The set of interpretations with this partial order is shown to be a complete lattice, and the greatest lower bound of any set of models is shown to be a model. Thus there exists a least model which happens to be the least Herbrand model and therefore the ground success set semantics.

Richer operational semantics are non-minimal models which can,
however, be effectively defined by fixpoint constructions. In Section 5 four different fixpoint operators are defined. One of these \( T_3 \), which corresponds to the computed answer substitution operational semantics and to the S-semantics in Falaschi et al. (1988, 1989), is the most primitive one in the sense that the others can easily be obtained from it. Correspondingly, in Section 6 various equivalence relations are shown, and, again, the one corresponding to the S-semantics turns out to be the finest one.

Finally, in Section 7 the fixpoint semantics are proved equivalent to the operational ones.

### 2. Operational Semantics

The reader is assumed to be familiar with the terminology of and the basic results in the semantics of logic programs (Lloyd, 1987; Apt, 1990).

Let the language \( L \) consist of:

- a finite set \( C \) of data constructors,
- a finite set \( P \) of predicate symbols
- a denumerable set \( V \) of variable symbols.

Let \( T \) be the set of terms built on \( C \) and \( V \). A substitution is a mapping \( \sigma : V \to T \) such that the set \( D(\sigma) = \{ X \mid \sigma(X) \neq X \} \) (domain of \( \sigma \)) is finite. \( e \) denotes the empty substitution. The composition \( \sigma \circ \sigma' \) of the substitutions \( \sigma \) and \( \sigma' \) is defined as the functional composition. The pre-ordering \( \preceq \) on substitutions is such that \( \sigma \preceq \sigma' \) if there exists \( \sigma' \) such that \( \sigma \sigma' = \sigma \).

An atom \( A \) is an object of the form \( p(t_1, \ldots, t_n) \), where \( p \in P \) and \( t_1, \ldots, t_n \in T \). The application of the substitution \( \sigma \) to the atom \( A \) is denoted by \( A \sigma \). We define \( A \preceq A' \) (\( A \) is more general than \( A' \)) if there exists \( \sigma \) such that \( A \sigma = A' \). The relation \( \preceq \) is a preorder. Let \( \approx \) be the associated equivalence relation (renaming).

A definite clause in the language \( L \) is a formula \( H \leftarrow B_1, \ldots, B_n \) (\( n \geq 0 \)), where \( H \) and the \( B_i \)'s are atoms, "\( \leftarrow \)" and "\( , \)" denote logic implication and conjunction respectively, and all variables are universally quantified. \( H \) is the head of the clause and \( B_1, \ldots, B_n \) is the body. Given a clause \( H \leftarrow B_1, \ldots, B_n \), a substitution \( \sigma \) is grounding if \( H \sigma, B_1 \sigma, \ldots, B_n \sigma \) are ground atoms. If the body is empty the clause is a unit clause. A HCL program on \( L \) is a finite set of definite clauses \( W = \{ C_1, \ldots, C_n \} \) in \( L \). A goal statement is a formula \( \leftarrow A_1, \ldots, A_m \), where each \( A_i \) is an atom in \( L \).

\( G \vdash^* \square \) denotes the SLD-refutation of a goal \( G \) with computed answer substitution \( \sigma \).

The aim of this paper is to relate the declarative (model-theoretic and
fixpoint) semantics to various observable (operational) properties of logic programs. In fact the standard declarative and operational semantics are not fully equivalent (as already noted in (Clark, 1979; Falaschi et al., 1988, 1989)). Moreover many different operational semantics can be defined according to different notions of observable properties (Gaifman and Shapiro, 1989a, b).

We will consider in the following some of these operational semantics and induced program equivalence relations, of which some can be found in Maher (1987).

2.1. Success Set Semantics

Let $W$ be a program. Its success set semantics is

$$O_1(W) = \{ A \mid A \text{ is ground and } \leftarrow A \leftarrow^{e} \square \}.$$

This is the standard operational semantics, which is equivalent to the least Herbrand model, as stated by the weak soundness and weak completeness theorems (van Emden and Kowalski, 1976).

However, the observational program equivalence based on the success set semantics ($W \approx_1 W'$ iff $O_1(W) = O_1(W')$) is too weak.

**Example 2.1.** Consider the programs $W_1$ and $W_2$ in the language $L_1$ defined by $C = \{a \backslash 0\}$ and $P = \{p \backslash 1, q \backslash 1\}$:

$$W_1 = \{p(a). q(X).\}$$

$$W_2 = \{p(a). q(a).\}.$$

$W_1 \approx_1 W_2$ since $O_1(W_1) = O_1(W_2) = \{p(a), q(a)\}$. However, the two programs have quite different operational behaviours. In fact, the goal $\leftarrow q(X)$ computes an empty answer substitution in $W_1$, while it computes the answer substitution $\{X \backslash a\}$ in $W_2$.

Moreover, if the program alphabet is extended to contain the constant $b$, then $W_1 \approx_1 W_2$ does not hold anymore, since the goal $\leftarrow q(b)$ is refutable in $W_1$, while it finitely fails in $W_2$.

A better equivalence relation can be based on a differently observable property, i.e., the set of atomic consequences or non-ground success set.

2.2. Non-ground Success Set Semantics

Let $W$ be a program. Its non-ground success set semantics is:

$$O_2(W) = \{ A \mid \leftarrow A \leftarrow^{e} \square \}$$
\( O_2(W) \) is the same as the set of atomic logic consequences of \( W \), as stated by the strong soundness and strong completeness theorems (Clark, 1979).

The corresponding equivalence relation \( \approx_2 \) (\( W \approx_2 W' \) iff \( O_2(W) = O_2(W') \)) is stronger than \( \approx_1 \). In fact, if we consider Example 2.1, \( W_1 \approx_2 W_2 \) does not hold, since

\[
O_2(W_1) = \{ p(a), q(X), q(a) \} \neq O_2(W_2) = \{ p(a), q(a) \}.
\]

However, \( \approx_2 \) is still too weak to characterize computed answer substitutions, as shown by the following example.

**Example 2.2.** Consider the programs \( W_1 \) and \( W_2 \) in the language \( L_1 \) of Example 2.1:

\[
W_1 = \{ p(a), q(X) \}
\]

\[
W_2 = \{ p(a), q(X), q(a) \}.
\]

\( W_1 \approx_2 W_2 \) since \( O_2(W_1) = O_2(W_2) = \{ p(a), q(a), q(X) \} \). However, the goal \( \leftarrow q(X) \) computes an answer substitution \( \{ X \backslash a \} \) in \( W_2 \) only.

This difference can be modeled by a stronger notion of observable property, i.e., computed answer substitutions.

### 2.3. Computed Answer Substitution Semantics

Let \( W \) be a program. Its computed answer substitution semantic is

\[
O_3(W) = \{ A \mid \exists p \in P, \\exists X_1, \ldots, X_n \text{ distinct variables in } V, \exists \partial, \leftarrow p(X_1, \ldots, X_n) \xrightarrow{a} \square, A = p(X_1, \ldots, X_n) \partial \}.
\]

The corresponding equivalence relation \( \approx_3 \) (\( W \approx_3 W' \) iff \( O_3(W) = O_3(W') \)) is stronger than \( \approx_2 \). In fact, if we consider Example 2.2, \( W_1 \approx_3 W_2 \) does not hold, since

\[
O_3(W_1) = \{ p(a), q(X) \} \neq O_3(W_2) = \{ p(a), q(a), q(X) \}.
\]

This operational semantics fully characterizes computed answer substitutions. Indeed, the semantics of a program \( W \) can be viewed as a
possibly infinite set of (unit) clauses and the computed answer substitutions can be obtained by executing the goal in the "program" $O_3(W)$.

$O_3(W)$ has a declarative counterpart in the $S$-semantics (see the soundness and completeness theorems in Section 7).

In the following we try to find a model-theoretic counterpart to the above different notions of observable properties.

3. Interpretations and Models

Two of the above introduced operational semantics ($O_2(W)$ and $O_3(W)$) are defined as sets of non-ground atoms. Therefore interpretations must contain non-ground atoms.

**Definition 3.1.** (Base). The Herbrand base $B$ is the quotient set of all the atoms with respect to $\approx$. The ordering induced by $\leq$ on $B$ will still be denoted by $\leq$. For the sake of simplicity, we will represent the equivalence class of an atom $A$ by $A$ itself.

**Definition 3.2.** (Set of Interpretations $\mathcal{I}$). An interpretation $I$ is any subset of $B$.

It is worth noting that $B$ (and therefore the set of interpretations $\mathcal{I}$) depends upon the language $L$ and not upon the program. Note also that the definitions of $O_2(W)$ and $O_3(W)$ should be given in terms of equivalence classes.

Let us now introduce some useful definitions of abstraction operators on interpretations, and then the notions of truth and model.

**Definition 3.3** (Abstraction Operators on Interpretations). Let $I$ be an interpretation.

- upward closure $[I] = \{A \in B | \exists A' \in I, A' \leq A\}$,
- ground atoms $[I] = \{A \in I | A$ is ground$\}$
- minimal elements $\text{Min}(I) = \{A \in I | \forall A' \in I \text{ if } A' \leq A \text{ then } A = A'\}$.

Let us introduce the notation $[I]$ as a shorthand for $\llbracket I \rrbracket$. Note that $[I]$ is the set of all the ground instances of atoms in $I$.

**Definition 3.4** (Truth). An atom $A$ or a definite clause $A \leftarrow B_1, \ldots, B_n$ is true in $I$ iff it is true (using the standard definition of truth in a Herbrand interpretation) in $[I]$. 
A few remarks are in order. A (possibly non-ground) atom \( A \) is true in \( I \) iff \( \{\bar{A}\} \subseteq [I] \). Note also that if \( A \in [I] \) then \( A \) is true in \( I \).

**Definition 3.5 (Models).** A model of a logic program \( W \) is any interpretation \( M \) in which all the clauses of \( W \) are true.

Note that for each standard Herbrand interpretation \( I \) there exist (possibly infinitely) many different interpretations \( I_1, I_2, \ldots \), such that \([I_1] = [I_2] = \cdots = I\). \( I \) is a standard Herbrand model if and only if \( I_1, I_2, \ldots \) are models. The \( I_i \)'s are therefore equivalent from the viewpoint of model theory. Nonetheless, as we show in the following, they are different, because they exhibit different computational properties.

**Proposition 3.6.** The (standard) Herbrand models are models (according to Definitions 3.4 and 3.5).

**Proof.** Immediate, from which \( I \) is a standard Herbrand interpretation \( I = [I] \).

**Corollary 3.7.** Every logic program has a model (according to Definitions 3.4 and 3.5).

**Proof.** Immediate from Proposition 3.6.

The following technical lemmas will be useful in later sections.

**Lemma 3.8.** For any interpretation \( I \), \( [I] = [\text{Min}(I)] \) and \( \text{Min}([I]) = \text{Min}(I) \) hold.

**Proof.** It is a straightforward consequence of Lemma 5.4 in (Lassez et al., 1987) which, according to our terminology, states that for any atom \( A \) there exists only a finite number of atoms \( A' \leq A \) up to variable renaming.

**Lemma 3.9.** Let \( A \) be any atom and \( I \) be any interpretation. \( A \) is true in \( [I] \) iff \( A \) is true in \( I \).

**Proof.** Straightforward, since \([I] = [I]\).

**Lemma 3.10.** For any interpretation \( I \), \( [I] \) is a model of a program \( W \) iff \( I \) is a model of \( W \).

**Proof.** This follows immediately from Lemma 3.9.

**Lemma 3.11.** Let \( A \) be any atom and \( I \) be any interpretation. \( A \) is true in \( \text{Min}(I) \) iff \( A \) is true in \( I \).
Proof. Immediate (by Lemma 3.8), since \( \Gamma \text{Min}(I) \models \Gamma I \).

Lemma 3.12. For any interpretation \( I \), \( \text{Min}(I) \) is a model of a program \( W \) iff \( I \) is a model of \( W \).

Proof. This follows immediately from Lemma 3.11.

In Section 7 we formally prove that the interpretations corresponding to the operational semantics defined in Section 2 are all models according to the above definition. Let us now give an example.

Example 3.13. Consider the program \( W \) in the language \( L_2 \), defined by \( C = \{ a, 0, f \} \) and \( P = \{ p, q \} \):

\[
W = \{ p(f(a)),
            p(X),
            q(a)
        \}
\]

\( O_1(W) = \{ q(a), p(a), p(f(a)), p(f(f(a))), ... \} \)

\( O_2(W) = \{ q(a), p(X), p(a), p(f(X)), p(f(a)), p(f(f(X))), p(f(f(a))), ... \} \)

\( O_3(W) = \{ q(a), p(X), p(f(a)) \} \).

It is easy to check that these interpretations are also models. However, the example shows that one relevant property of standard Herbrand models does not hold anymore; namely, the intersection of a set of models is not always a model. In this case the intersection of \( O_1(W), O_2(W), O_3(W) \) is \( \{ q(a), p(f(a)) \} \), and it is not a model.

Therefore in general there exists no least model with respect to set inclusion. Hence we try to define a partial order on interpretations to restore the model intersection property, thus making it possible to define a unique model as the model-theoretic semantics. This partial order also allows us to compare various interesting models.

4. Model-Theoretic Semantics

The absence of the least model with respect to set inclusion can easily be explained by noting that set inclusion does not adequately reflect the property of non-ground atoms of being representatives of all their ground instances. This is the first property to be considered in the definition of the new partial order relation. An additional property is related to the ability of modeling computed answer substitutions. This is shown by the fact that the interpretation \( \{ p(X), p(a) \} \) has more information than the interpreta-
tion \( \{ p(X) \} \), even if it has the same ground instances. The two properties are embedded in the following definition.

**Definition 4.1.** Let \( I_1, I_2 \) be interpretations. We define:

- \( I_1 \leq I_2 \) iff \( \forall A \in I_1, \exists A_2 \in I_2 \) such that \( A_2 \leq A_1 \).
- \( I_1 \sqsubseteq I_2 \) iff \( (I_1 \leq I_2) \) and \( (I_2 \leq I_1) \) implies \( I_1 \sqsubseteq I_2 \).

The intuitive meaning of the above defined relations is the following: \( I_1 \leq I_2 \) means that every atom verified by \( I_1 \) is also verified by \( I_2 \) (\( I_2 \) contains more positive information). Note that \( \leq \), with some abuse of notation, has different meanings for atoms and interpretations. \( I_1 \sqsubseteq I_2 \) means either that \( I_2 \) strictly contains more positive information than \( I_1 \) or (if the amount of positive information is the same) that \( I_1 \) expresses it by fewer elements than \( I_2 \) (\( I_2 \) is more redundant).

The following two lemmas have a straightforward proof.

**Lemma 4.2 (\( \leq \) is a Preorder).** The relation \( \leq \) of Definition 4.1 is a preorder.

**Lemma 4.3.** (a) If \( I_1 \leq I_2 \) then \( I_1 \leq I_2 \).

(b) \( I_1 \sqsubseteq I_2 \) iff \( \forall I_1 \sqsubseteq \forall I_2 \).

**Proposition 4.4 (\( \sqsubseteq \) on Interpretations Is an Ordering).** The relation \( \sqsubseteq \) of Definition 4.1 is an ordering.

**Proof.** (reflexivity) \( I \sqsubseteq I \). In fact, \( I \leq I \) (by Lemma 4.2) and \( I \sqsubseteq I \).

(antisymmetry) Assume \( I_1 \sqsubseteq I_2 \) and \( I_2 \sqsubseteq I_1 \). Then, by definition, \( I_1 \leq I_2 \), and \( I_2 \leq I_1 \). Then, again by definition, we obtain both \( I_1 \leq I_2 \) and \( I_2 \leq I_1 \). Therefore \( I_1 = I_2 \).

(transitivity) Assume \( I_1 \sqsubseteq I_2 \) and \( I_2 \sqsubseteq I_3 \). Then, by definition, \( I_1 \leq I_2 \), and \( I_2 \leq I_3 \). By Lemma 4.2, \( I_1 \leq I_3 \). Assume now \( I_3 \sqsubseteq I_1 \). Since \( I_1 \leq I_2 \), by Lemma 4.2, \( I_3 \leq I_2 \). Since \( I_2 \leq I_3 \), \( I_2 \leq I_3 \), by definition. By applying the same argument to \( I_2 \) and \( I_1 \), we obtain \( I_1 \leq I_2 \). Therefore \( I_1 \leq I_3 \).

**Proposition 4.5.** If \( I_1 \leq I_2 \), then \( I_1 \sqsubseteq I_2 \).

**Proof.** Immediate, since, by Lemma 4.3(a), \( I_1 \leq I_2 \) implies \( I_1 \leq I_2 \).

**Example 4.6.** Let the base \( B \) be the set \( \{ p(X), p(a), p(b) \} \). The set \( \mathcal{I} \) of interpretations is ordered as shown in Fig. 1, where a directed arc from \( I_1 \) to \( I_2 \) denotes \( I_1 \sqsubseteq I_2 \).
Now we discuss the properties of the relation \( \subseteq \) on interpretations. We show that the set of interpretations under the relation \( \subseteq \) is a complete lattice.

**Definition 4.7.** Let \( \Gamma \) be a set of interpretations. We introduce the following notations:

- \( \forall \Gamma = \bigcup_{I \in \Gamma} I \)
- \( \text{Min}(\Gamma) = \text{Min}(\forall \Gamma) \)
- \( \bigcup \Gamma = \text{Min}(\Gamma) \cup \forall \{ I \in \Gamma \mid \text{Min}(\Gamma) \subseteq I \} \).

Note that \( \text{Min}(\Gamma) = \text{Min}(\bigcup \Gamma) \).

**Proposition 4.8.** For any set \( \Gamma \) of interpretations there exists the least upper bound of \( \Gamma \), \( \text{lub}(\Gamma) \), and \( \text{lub}(\Gamma) = \bigcup \Gamma \) holds.

**Proof.**

1. \( (\bigcup \Gamma \) is an upper bound of \( \Gamma \)) If \( I \) is an element of \( \Gamma \), then \( \Gamma \text{Min}(I) \subseteq \Gamma \text{Min}(\bigcup \Gamma) = \Gamma \text{Min}(\bigcup \Gamma) \). Therefore, by Lemma 3.8, \( \bigcup \Gamma \subseteq \bigcup \Gamma \) holds. Then, by Lemma 4.3(b), \( I \subseteq \bigcup \Gamma \). Moreover, if \( \bigcup \Gamma \subseteq I \), then, by Lemma 4.3(b), \( \bigcup \Gamma \subseteq \bigcup \Gamma \). Then \( \text{Min}(\Gamma) = \text{Min}(\bigcup \Gamma) \subseteq I \). Therefore, by Definition of \( \bigcup \Gamma \), \( I \subseteq \bigcup \Gamma \).

2. \( (\bigcup \Gamma \) is the least upper bound of \( \Gamma \)) Let \( H \) be an upper bound of \( \Gamma \). Since, for every \( I \in \Gamma \) (by Lemma 4.3(b)) \( \Gamma \subseteq \Gamma H \), then \( \bigcup \Gamma \subseteq \Gamma H \);
i.e., \( \bigcup \Gamma \subseteq H \). Assume now \( H \subseteq \bigcup \Gamma \); i.e., \( \bigcap H \subseteq \bigcup \Gamma \). Then \( \bigcap H = \bigcup \Gamma \) and therefore \( \text{Min}(\Gamma) = \text{Min}(\bigcup \Gamma) \subseteq H \). Moreover, for any \( I \in \Gamma \) such that \( \text{Min}(\Gamma) \subseteq I \), \( \bigcap H \subseteq I \) and therefore (since \( I \subseteq H \)) \( I \subseteq H \). Then \( \forall \{I \in \Gamma \mid \text{Min}(\Gamma) \subseteq I \} \subseteq H \) and therefore \( \bigcup \Gamma \subseteq H \) holds.

**Theorem 4.9.** The set of all the interpretations \( \mathcal{I} \) with the ordering \( \subseteq \) is a complete lattice. \( \mathcal{B} \) is the top element and \( \emptyset \) is the bottom element.

**Proof.** For any set \( \Gamma \) of interpretations, the existence of its least upper bound is ensured by Proposition 4.8. The greatest lower bound of \( \Gamma \) is then given by

\[
\text{glb}(\Gamma) = \text{lub}(\{I \in \mathcal{I} \mid \forall I' \in \Gamma, I \subseteq I'\}).
\]

An important property of the standard Herbrand models (that allows us to show the existence of the least one) is the model intersection property, which states that the intersection of a set of models is still a model. The following proposition generalizes this result.

**Proposition 4.10.** Let \( \mathcal{M} \) be a set of models of a program \( W \). Then \( \text{glb}(\mathcal{M}) \) is a model of \( W \).

**Proof.** Let \( A \leftrightarrow B_1, \ldots, B_n \) be a clause of \( W \). Consider a substitution \( \theta \). Assume that \( B_1\theta, \ldots, B_n\theta \) are true in \( \text{glb}(\mathcal{M}) \). Then \( \{B_1\theta, \ldots, B_n\theta\} \subseteq \bigcap \text{glb}(\mathcal{M}) \). By definition of glb, for any \( I \in \mathcal{M} \), \( \text{glb}(\mathcal{M}) \subseteq I \) holds. Therefore \( \bigcap \text{glb}(\mathcal{M}) \subseteq I \) and then \( \{B_1\theta, \ldots, B_n\theta\} \subseteq I \). Since \( A \leftrightarrow B_1, \ldots, B_n \) is true in \( I \), \( \{A\theta\} \subseteq I \), by definition. This implies, by Proposition 4.5, \( \{A\theta\} \subseteq \bigcap I \). Hence, \( \{A\theta\} \) is a lower bound of \( \mathcal{M} \), and therefore \( \{A\theta\} \subseteq \text{glb}(\mathcal{M}) \). By definition, \( \bigcap \{A\theta\} \subseteq \text{glb}(\mathcal{M}) \) and therefore \( \bigcap \{A\theta\} = \bigcap \{A\theta\} \subseteq \text{glb}(\mathcal{M}) \).

**Corollary 4.11.** The set of all the models of a program \( W \) with the ordering \( \subseteq \) is a complete lattice.

We are now in the position to formally define the model-theoretic semantics.

**Definition 4.12.** Let \( W \) be a program. Its model-theoretic semantics is the greatest lower bound of the set of its models; i.e., \( M_1(W) = \text{glb}(\{I \in \mathcal{I} \mid I \text{ is a model of } W\}) \).

In the following we show that the above defined model-theoretic semantics is exactly the standard least Herbrand model. This fact justifies our choice of the ordering relation.
LEMMA 4.13. For any model \( I \) there exists a standard Herbrand model \( I' \) such that \( I' \subseteq I \).

Proof. Define \( I' = [I] \). Then \( I' \) is a standard Herbrand model (immediate). We show now that \( I' \subseteq I \). By definition, \([I'] = [I] \subseteq [I]\). Assume now \([I'] \subseteq [I']\). Since \([I'] = I', [I] \subseteq I'\), and therefore \( I \) is ground (\( I = [I] \)). Hence \( I' = [I] = [I] = I \). \( \blacksquare \)

COROLLARY 4.14. There exists a retraction \( \langle \Phi, \Psi \rangle \), i.e., a pair of functions \( \langle \Phi, \Psi \rangle \) such that \( \Phi \) is injective and \( \Psi \) is surjective, from the set of standard Herbrand models to the set of models such that \( \Phi(\Psi(I)) \subseteq I \).

Proof. Let \( I, I' \) range on the set of models and on the set of Herbrand models respectively. Define \( \Phi \) and \( \Psi \) as follows: \( \Phi(I') = I' \) and \( \Psi(I) = [I] \). By Proposition 3.6 this is a correct definition of retraction. The corollary follows from Lemma 4.13. \( \blacksquare \)

THEOREM 4.15. The least standard Herbrand model is the least model.

Proof. Note that the standard Herbrand models are ordered by set inclusion, then apply Propositions 4.5 and 3.6 and Lemma 4.13. \( \blacksquare \)

We now consider the relation between our models and C-models and S-models, which were defined in Falaschi et al. (1989) on the same set of interpretations. C-models and S-models were intended to capture specific operational properties, from a model-theoretic point of view. In both cases, an ad hoc notion of truth was considered. Let us recall the definitions of C-truth and S-truth from Falaschi et al. (1989).

DEFINITION 4.16 (C-Truth). Let \( I \) be a C-interpretation, i.e., a subset of \( B \) such that \( I = [I] \).

- A (possibly non-ground) atom \( A \) is C-true in \( I \) iff \( A \in [I] \).
- A definite clause \( A \leftarrow B_1, ..., B_n \) is C-true in \( I \) iff for each instance \( A\delta \leftarrow B_1\delta, ..., B_n\delta \), if \( B_1\delta, ..., B_n\delta \) are C-true in \( I \), then \( A\delta \) is C-true in \( I \).

DEFINITION 4.17 (S-Truth). Let \( I \) be an S-interpretation, i.e., any subset of \( B \).

- A (possibly non-ground) atom \( A \) is S-true in \( I \) iff \( \exists A' \in \[I\], A' \subseteq A \).
- A definite clause \( A \leftarrow B_1, ..., B_n \) is S-true in \( I \) iff for each \( B_1', ..., B_n' \in I \) if \( \delta = \text{mgu}((B_1, ..., B_n), (B_1', ..., B_n')) \), then \( A\delta \in I \).

C-models and S-models are defined in the obvious way.
Theorem 4.18. Every C-model is a model.

Proof. (atomic case) For each atom A, A is C-true in a C-interpretation I iff A ∈ I = [I]. Then [A] ⊆ [I], and then [[A]] ⊆ [I].

(non-atomic case) Consider a clause A ← B₁, ..., Bₙ and a C-interpretation I. For each substitution θ, if B₁θ, ..., Bₙθ are true in I, i.e., if ([[B₁θ, ..., Bₙθ]]) ⊆ [I]), we must show that [[Aθ]] ⊆ [I]. Consider a substitution γ such that Aθγ is ground. Then [[B₁θγ, ..., Bₙθγ]] ⊆ [I]. Let η be a substitution such that B₁θηγ, ..., Bₙθηγ are ground. Therefore B₁θηγ, ..., Bₙθηγ ∈ [I]. Since I is a C-interpretation, Aθγ = Aθηγ ∈ [I] for every γ such that Aθγ is ground. Therefore [[Aθ]] ⊆ [I].

The following results are proved in Falaschi et al. (1989).

Proposition 4.19 (Falaschi et al., 1989). Let I ∈ C. If I is a C-model then I is an S-model.

Proposition 4.20 (Falaschi et al., 1989). If I is an S-model, then [I] is a C-model.

Proposition 4.21 (Falaschi et al., 1989). For every program W there exist both a least S-model and a least C-model.

Proposition 4.22 (Falaschi et al., 1989). If I is an S-model for a program W then [I] is a Herbrand model of W.

Proposition 4.23 (Falaschi et al., 1989). For every program W, Mₛ(W) = [Mₛ(W)]

Theorem 4.24. Every S-model is a model.

Proof. By Proposition 4.20, if I is an S-model, then [I] is a C-model and therefore it is a model by Theorem 4.18. By Lemma 3.10 I is also a model.

Let W be a program. Let us define the following subsets of the set of interpretations C:

\[ \mathcal{M}_H = \{ I ∈ C \mid I \text{ is a Herbrand model of } W \} \]
\[ \mathcal{M}_s = \{ I ∈ C \mid I \text{ is a model of } W \} \]
\[ \mathcal{M}_c = \{ I ∈ C \mid I \text{ is a C-model of } W \} \]
\[ \mathcal{M}_s = \{ I ∈ C \mid I \text{ is an S-model of } W \} \]
Fig. 2. Models, Herbrand models, S-models, and C-models.

The following facts are true: $\mathcal{M}_H \subseteq \mathcal{M}_2$ (immediate, since if $I \in \mathcal{M}_H$ then $I = \overline{\cap I}$), $\mathcal{M}_2 \subseteq \mathcal{M}_1$ (Theorem 4.18), $\mathcal{M}_3 \subseteq \mathcal{M}_1$ (Theorem 4.24), $\mathcal{M}_2 \subseteq \mathcal{M}_3$ (Proposition 4.19).

Figure 2 shows the relations among the various sets and models.

**Definition 4.25.** Let $W$ be a program. We define the following interpretations:

- $M_2(W) = \text{glb}(\mathcal{M}_2),$
- $M_3(W) = \text{glb}(\mathcal{M}_3),$
- $M_4(W) = \text{Min}(M_3(W)).$

**Lemma 4.26.** For every program $W$, $M_i(W)$ is the least element of $\mathcal{M}_i$, $i = 1, 2, 3$.

**Proof.** ($i = 1$) This follows immediately from Theorem 4.15.

($i = 2, 3$) $\cap M_i \in \mathcal{M}_i$ by Proposition 4.21. Moreover, $\cap M_i$ is a lower bound of $\mathcal{M}_i$, by Proposition 4.5. Therefore $\cap M_i = \text{glb}(\mathcal{M}_i)$.

**Theorem 4.27.** For every program $W$, $M_2(W)$, $M_3(W)$, and $M_4(W)$ are models.

**Proof.** For $M_2(W)$ and $M_3(W)$, this derives from Lemma 4.26 and Theorems 4.18 and 4.24. For $M_4(W)$ it follows immediately from Lemma 3.12.
PROPOSITION 4.28. The following relations between the various models hold:

- \( M_1(W) = \downarrow M_3(W) \downarrow \)
- \( M_2(W) = \uparrow M_4(W) \uparrow \)
- \( M_4(W) = \text{Min}(M_3(W)) \).

Proof. First statement:

\( A \in M_1(W) \)

iff \( W \models A \) (by Theorem 4.15)

iff \( A \) is S-true in \( M_3(W) \) (by Theorem 5.7 in Falaschi et al., 1989)

iff \( A \in [M_3(W)] \) (since \( A \) is ground).

The second statement is Proposition 4.23.
The third statement is by definition.

COROLLARY 4.29. The following relations between the various models also hold:

- \( M_1(W) = \downarrow M_2(W) \downarrow \)
- \( M_2(W) = \uparrow M_4(W) \uparrow \)
- \( M_4(W) = \text{Min}(M_2(W)) \).

Proof. This derives immediately from Proposition 4.28 and Lemma 3.8.

The relations among the various models are shown in Fig. 3.

THEOREM 4.30. For every program \( W \),

\[ M_1(W) \subseteq M_4(W) \subseteq M_3(W) \subseteq M_2(W). \]

Proof. The first relation is immediate since \( M_1(W) \) is the glb of the set of all the models. For the second relation note that \( M_4(W) = \)

![Diagram showing models and their mappings.](image)

Fig. 3. Models and their mappings.
\[
\text{Min}(M_3(W)) \subseteq M_3(W). \text{ Then apply Proposition 4.5. The third relation derives from Proposition 4.28 and from Proposition 4.5.} \]

As we show in later sections, \( M_1(W), M_2(W), \) and \( M_3(W) \) are equivalent to \( O_1(W), O_2(W), \) and \( O_3(W). \) \( M_4(W) \) was originally proposed in Gaifman and Shapiro (1989a), and will be proved to give the same observational equivalence as \( M_3(W). \) Figure 3 shows that \( M_3(W) \) (the \( S \)-semantics) is the model which has the richest information content. In fact, the other models can be obtained by applying suitable abstraction operators, and not vice versa. Note that the ordering relation among the models, as stated in Theorem 4.30, is not directly related to the information content. In particular, \( M_3(W), \) which has the richest information content, is a non-least model.

5. Fixpoint Semantics

In this section we show that the four models that were introduced in Section 4 can all be obtained as least fixpoints of transformations on interpretations.

**Definition 5.1 (Transformations on \( \mathcal{F} \)).**

1. \( T_1(I) = \{ A \in \mathcal{B} | A \text{ is ground}, \)
   \[
   \exists A' \leftarrow B_1, ..., B_n \in W, \\
   \exists \emptyset \text{ grounding,} \\
   B_1 \emptyset, ..., B_n \emptyset \in [I], \\
   A = A' \emptyset \}
   \]

2. \( T_2(I) = \{ A \in \mathcal{B} | \exists A' \leftarrow B_1, ..., B_n \in W, \)
   \[
   \exists \emptyset, \\
   B_1 \emptyset, ..., B_n \emptyset \in [I], \\
   A = A' \emptyset \}
   \]

3. \( T_3(I) = \{ A \in \mathcal{B} | \exists C = A' \leftarrow B_1, ..., B_n \in W, \)
   \[
   \exists B'_1, ..., B'_n \text{ variants of atoms in } I, \\
   \text{with no variables in common} \\
   \text{with } C \text{ and with each other,} \\
   \exists \emptyset = \text{mgu}((B_1, ..., B_n), (B'_1, ..., B'_n)), \\
   A = A' \emptyset \}
   \]

4. \( T_4(I) = \text{Min}(T_2(I)). \)
Lemma 5.2.

- \( T_1(I) = T_1([I]) \)
- \( T_2(I) = T_2([I]) \).

Proof. Immediate.

Proposition 5.3. The following relations among the various transformations hold:

- \( T_1(I) = [T_3(I)] \)
- \( T_2(I) = [T_3(I)] \)
- \( T_4(I) = \min(T_3(I)) \).

Proof. The first statement follows from Lemma 5.2 and Proposition 6.12(b) in Falaschi et al. (1989). Analogously, the second statement follows from Lemma 5.2 and Proposition 6.7 in Falaschi et al. (1989).

The third statement is immediate by Lemma 3.8.

It is worth noting that \( T_3 \) is the most basic transformation, as was the case for model \( M_3 \). The models we are interested in are the least fixpoints of the above defined transformations. This can be shown by first proving the continuity properties.

Proposition 5.4 (Monotonicity and Continuity of \( T_1 \)). \( T_1 \) is monotonic and continuous in the complete lattice \( \langle \mathcal{F}, \subseteq \rangle \).

Proof. (monotonicity) Let \( I \subseteq I' \). Then, \( [I] \subseteq [I'] \) and therefore \( T_1(I) \subseteq T_1(I') \). By Proposition 4.5 we derive \( T_1(I) \subseteq T_1(I') \).

(continuity) \( \lub_{I \in I} T_1(I) \subseteq T_1(\lub(I)) \) follows by monotonicity. Therefore it is sufficient to show that for any chain \( I, T_1(\lub(I)) \subseteq \lub_{I \in I} T_1(I) \) holds. It is easy to see that \( T_1(\bigvee I) \subseteq \bigcup_{I \in I} T_1(I) \). Indeed, \( \{[I] | I \in I \} \) is a chain ordered by set inclusion, \( T_1(I) = T_1([I]) \), and \( T_1 \) is continuous with respect to set inclusion. Moreover, since \( T_1(I) = \bigcup_{I \in I} T_1(I) \), \( \bigcup_{I \in I} T_1(I) \) is ground. Note that if \( I \) is ground \( \min(I) = I \), and the \( \lub_{I \in I} I \), when all the \( I \)'s are ground, is \( \min(I) \cup \bigvee \{J \in I | \min(J) \subseteq J \} = \bigvee I \). Thus \( \bigcup_{I \in I} T_1(I) = \lub_{I \in I} T_1(I) \) holds. Finally, note that \( \lub(I) \subseteq \bigvee I \). Therefore, \( T_1(\lub(I)) \subseteq T_1(\bigvee I) = \bigcup_{I \in I} T_1(I) \). By Proposition 4.5, \( T_1(\lub(I)) \subseteq \lub_{I \in I} T_1(I) \).

Proposition 5.5 (Monotonicity and Continuity of \( T_2, T_3 \) (Falaschi et al., 1989)). \( T_2 \) and \( T_3 \) are monotonic and continuous in the complete lattice \( \langle \mathcal{F}, \subseteq \rangle \).

Proposition 5.6 (Monotonicity and Continuity of \( T_4 \)). \( T_4 \) is monotonic and continuous in the complete lattice \( \langle \mathcal{F}, \subseteq \rangle \).
Proof. (monotonicity) By definition and Lemma 3.8, $T_4(I) = \text{Min}(T_3(I))$. $T_2$ can be shown to be monotonic in the complete lattice $\langle \mathcal{F}, \subseteq \rangle$ by exactly the same arguments used in Proposition 5.4. Let $I \subseteq I'$. Then $[T_4(I)] \subseteq [T_3(I)] \subseteq [T_2(I')] = [T_4(I')]$. Assume now that $[T_4(I')] \subseteq [T_4(I)]$. Then $[\text{Min}(T_3(I'))] \subseteq [\text{Min}(T_2(I))]$. By Lemma 3.8, $T_2(I') \subseteq T_2(I)$. Hence $T_2(I') = T_2(I)$, and therefore $T_4(I) = \text{Min}(T_3(I)) = \text{Min}(T_2(I')) = T_4(I')$.

(continuity) $T_2$ is continuous in the complete lattice $\langle \mathcal{F}, \subseteq \rangle$ and $T_2(I) = T_2([I])$. Then, for any chain $\Gamma$, $T_2(\bigvee \Gamma) \subseteq \bigcup_{\Gamma \in \Gamma} T_2(I)$, and therefore $[T_4(\text{lub}(\Gamma))] = [T_3(\text{lub}(\Gamma))] = [T_3([\text{lub}(\Gamma)])] = T_2([\text{lub}(\Gamma)]) \subseteq \bigcup_{\Gamma \in \Gamma} T_2(I)$. Assume now that $[\text{lub}_{\Gamma \in \Gamma} T_2(I)] \subseteq [T_4(\text{lub}(\Gamma))]$. Then $\text{Min}(T_4(\text{lub}(\Gamma))) = \text{Min}(\text{lub}_{\Gamma \in \Gamma} T_4(I))$. Since $\text{Min}(T_4(\text{lub}(\Gamma))) = T_4(\text{lub}(\Gamma))$, $T_4(\text{lub}(\Gamma)) \subseteq \text{lub}_{\Gamma \in \Gamma} T_4(I)$. By Proposition 4.5, $T_4(\text{lub}(\Gamma)) \subseteq \text{lub}_{\Gamma \in \Gamma} T_4(I)$.

The continuity of $T_1$ and $T_4$ ensures the existence of their least fixpoints in the complete lattice $\langle \mathcal{F}, \subseteq \rangle$. They are the least upper bounds of the chains obtained by iterating $T_1$ and $T_4$ up from $\emptyset$. In the case of $T_2$ and $T_3$, they are continuous in the complete lattice $\langle \mathcal{F}, \subseteq \rangle$ only. However, as we will show in the following, there exist their least fixpoints in the complete lattice $\langle \mathcal{F}, \subseteq \rangle$, which are the unions of the chains obtained by iterating $T_2$ and $T_3$ up from $\emptyset$.

**Definition 5.7.** For $i = 1, 2, 3, 4$ define:

- $T_i \uparrow 0 = \emptyset$
- $T_i \uparrow n + 1 = T_i(T_i \uparrow n)$
- $T_i \uparrow \omega = \begin{cases} \text{lub}_n (T_i \uparrow n) & \text{for } i = 1 \text{ and } i = 4, \\ \bigcup_{n \geq 0} (T_i \uparrow n) & \text{for } i = 2 \text{ and } i = 3. \end{cases}$

**Theorem 5.8.** For $i = 1, 2, 3, 4$, $T_i \uparrow \omega$ is the least fixpoint of $T_i$ in the complete lattice $\langle \mathcal{F}, \subseteq \rangle$.

Proof. For $i = 1, 4$ this follows from Propositions 5.4 and 5.6. Now we consider the cases $i = 2, 3$.

(a) $(\bigcup_{n \geq 0} (T_i \uparrow n))$ is a fixpoint of $T_i$. $T_i$ is continuous with respect to set inclusion (Proposition 5.5). Therefore $\bigcup_{n \geq 0} (T_i \uparrow n)$ is a fixpoint of $T_i$.

(b) $(\bigcup_{n \geq 0} (T_i \uparrow n))$ is the least fixpoint of $T_i$. By continuity of $T_i$, $\bigcup_{n \geq 0} (T_i \uparrow n)$ is the least fixpoint with respect to set inclusion. Then apply Proposition 4.5.
A few remarks are in order to explain the need to resort to the complete lattice \( \langle \mathcal{S}, \subseteq \rangle \) in the case of \( T_2 \) and \( T_3 \).

\( T_3 \) is not monotonic in the complete lattice \( \langle \mathcal{S}, \subseteq \rangle \) (and therefore it is not continuous), as shown by the following example.

**Example 5.9.** Consider the program \( W \) in the language defined by \( C = \{ a \uparrow 0 \} \) and \( P = \{ p \uparrow 1 \} \):

\[
W = \{ p(X), \quad p(X) \leftarrow p(X) \}. 
\]

Let \( I_1 = \{ p(a) \}, \quad I_2 = \{ p(X) \} \). Then \( I_1 \subseteq I_2 \), while \( T_3(I_1) = \{ p(a), p(X) \} \not\subseteq T_3(I_2) = \{ p(X) \} \).

In general, \( \bigcup_n (T_3 \uparrow n) \) is different from \( \text{lub}_n(T_3 \uparrow n) \), as shown by the following example; i.e., the least fixpoint is not the least upper bound. The relation \( \text{lub}_n(T_3 \uparrow n) \subseteq \bigcup_n (T_3 \uparrow n) \) holds in general.

**Example 5.10.** Let \( W \) be the program

\[
\{ p(0, X), \quad p(s(Y), s(X)) \leftarrow p(Y, X), \quad p(Y, X) \leftarrow p(Y, s(X)) \}. 
\]

in the language \( L \) defined by \( C = \{ 0 \uparrow 0, s \uparrow 1 \} \) and \( P = \{ p \uparrow 2 \} \),

\[
T_3 \uparrow 1 = \{ p(0, X) \}, \\
T_3 \uparrow 2 = \{ p(0, X), p(s(0), s(X)) \}, \\
T_3 \uparrow 3 = \{ p(0, X), p(s(0), s(X)), p(s^2(0), s^2(X)), p(s(0), X) \}, \\
\ldots .
\]

Then \( \text{lub}_n(T_3 \uparrow n) \) is \( \{ p(0, X), p(s(0), X), p(s^2(0), X), \ldots \} \) and does not contain, for instance, \( p(s(0), s(X)) \in T_3 \uparrow 2 \).

Also in the case of \( T_2 \) the least upper bound is not a fixpoint, as shown by the following example.

**Example 5.11.** Let \( W \) be the program

\[
\{ p(0, X), \quad p(s(Y), X) \leftarrow p(Y, X) \}. 
\]
in the language $L$ defined by $C = \{0 \setminus 0, s \setminus 1\}$ and $P = \{p \setminus 2\}$,

$$T_2 \uparrow 1 = \{ p(0, t) \mid t \in T \},$$

$$\ldots$$

$$T_2 \uparrow n = \{ p(s^k(0), t) \mid k \leq n - 1, t \in T \}.$$

Because of the definition of $\sqsubseteq$ and Definition 4.7

$$\text{lub}_n(T_2 \uparrow n) = \{ p(s^k(0), X) \mid k \in \omega \}$$

which is not a fixpoint since

$$T_2(\{ p(s^k(0), X) \mid k \in \omega \}) = \{ p(s^k(0), t) \mid k \in \omega, t \in T \}.$$

The following lemmas are needed to prove the relation between models and fixpoints.

**Lemma 5.12 (Falaschi et al., 1989).** An S-interpretation $I$ is an $S$-model iff $T_3(I) \sqsubseteq I$.

**Lemma 5.13 (Falaschi et al., 1989).** A C-interpretation $I$ is a C-model iff $T_2(I) \subseteq I$.

**Theorem 5.14.** For $i = 1, 2, 3, 4$ if $I$ is a fixpoint of $T_i$, then $I$ is a model of $W$.

**Proof.** ($i = 2, 3$) This derives from Lemmas 5.12 and 5.13 and Theorems 4.18 and 4.24 (S-models and C-models are models).

($i = 1$) If $I$ is a fixpoint of $T_1$, then $I = \downarrow I$ and $I$ is a standard Herbrand model of $W$ (van Emden and Kowalski, 1976). The result can now be obtained by applying Proposition 3.6.

($i = 4$) Let $I$ be a fixpoint; i.e., $T_4(I) = I$. Under the hypothesis that $I$ is a fixpoint of $T_4$, we can always express $I$ as $\text{Min}(\Gamma J)$ for some $J$ ($T_4(I) = \text{Min}(T_2(I)) = \text{Min}(\Gamma T_2(I)) = I$). Thus $T_4(\text{Min}(\Gamma J)) = \text{Min}(\Gamma J)$.

By definition of $T_4$, $\text{Min}(T_2(\text{Min}(\Gamma J))) = \text{Min}(\Gamma J)$.

Since $T_2(\text{Min}(I)) = T_2(I)$ (see Lemma 3.8 and definition of $T_2$),

$\text{Min}(T_2(\Gamma J)) = \text{Min}(\Gamma J)$.

Since both the arguments of $\text{Min}$ are closed under the Up operation, we can deduce that $T_2(\Gamma J) = \Gamma J$; i.e., $\Gamma J$ is a fixpoint of $T_2$, and thus it is a model. Finally, by Lemma 3.12 also $\text{Min}(\Gamma J) = I$ is a model.

We can now give the formal definition of the fixpoint semantics.

**Definition 5.15 (Fixpoint Semantics).** For $i = 1, 2, 3, 4$, define $F_i(W) = T_i \uparrow \omega$. 

Proposition 5.17 shows the relation between the $F_i$'s. Let us first prove a technical lemma.

**Lemma 5.16.** \( \operatorname{Min}(T_3(\operatorname{Min}(I))) = \operatorname{Min}(T_3(I)) \).

**Proof.** \( (\subseteq) \) Straightforward, since \( \operatorname{Min}(I) \subseteq I \) and \( T_3 \) is monotonic w.r.t. \( \subseteq \).

\( (\supseteq) \) \( A \in \operatorname{Min}(T_3(I)) \) implies \( \exists A' \leftarrow B_1, ..., B_n \), where \( B_1', ..., B_n' \in I \), \( \mathcal{G} = \operatorname{mgu}((B_1, ..., B_n), (B_1', ..., B_n')) \) and \( A = A' \mathcal{G} \).

There exist \( B_1'', ..., B_n'' \in \operatorname{Min}(I) \) renamed apart from \( B_1, ..., B_n \) and from each other, where \( (B_1'', ..., B_n'') \leq (B_1', ..., B_n') \).

Thus there exists \( \mathcal{G}'' = \operatorname{mgu}((B_1'', ..., B_n''), (B_1, ..., B_n)) \) and \( \mathcal{G}'' \mathcal{G} \leq \mathcal{G} \).

Hence \( A'' = A' \mathcal{G}'' \leq A \in \operatorname{Min}(T_3(I)) \). But \( A'' \in T_3(\operatorname{Min}(I)) \subseteq T_3(I) \) implies \( A'' = A' \).

**Proposition 5.17.** Let \( W \) be a program

(a) \( F_1(W) = [F_3(W)] \)

(b) \( F_2(W) = [F_3(W)] \)

(c) \( F_4(W) = \operatorname{Min}(F_3(W)) \).

**Proof.** (a) We prove by induction that \( T_1 \uparrow n = [T_3 \uparrow n] \).

\( n = 1 \) This is derived by Proposition 5.3.

\( n = k + 1 \)

\[ T_1 \uparrow k + 1 = T_1(T_1 \uparrow k) \quad \text{(by the inductive hypothesis)} \]

\[ = T_1([T_3 \uparrow k]) \quad \text{(by Proposition 5.3)} \]

\[ = T_1(T_3 \uparrow k) \quad \text{(by Lemma 5.2)} \]

\[ = [T_3(T_3 \uparrow k)] \]

\[ = [T_3 \uparrow k + 1]. \]

We can now prove that \( F_1(W) = [F_3(W)] \).

\[ F_1(W) = \operatorname{lub}_{n \in \omega}(T_1 \uparrow n) \quad \text{(since \( T_1 \uparrow n = [T_3 \uparrow n] \))} \]

\[ = \bigcup_{n \in \omega} T_1 \uparrow n \quad \text{(by the previous proof)} \]

\[ = \bigcup_{n \in \omega} [T_3 \uparrow n] \quad \text{(since \( \bigcup_{i \in I} [i] = [\bigvee_i i] \))} \]

\[ = \left[ \bigcup_{n \in \omega} T_3 \uparrow n \right] \]

\[ = [F_3(W)]. \]
(b) The proof is similar to the one of case (a).

(c) We prove by induction that $T_4 \uparrow n = \text{Min}(T_3 \uparrow n)$.

\((n = 1)\) This is derived by Proposition 5.3.

\((n = k + 1)\)

\[
T_4 \uparrow k + 1 = T_4(T_4 \uparrow k)
\]
(by the inductive hypothesis)

\[
= T_4(\text{Min}(T_3 \uparrow k))
\]
(by Proposition 5.3)

\[
= \text{Min}(T_3(\text{Min}(T_3 \uparrow k)))
\]
(by Lemma 5.16)

\[
= \text{Min}(T_3 \uparrow k + 1)).
\]

We can now prove that $F_4(W) = \text{Min}(F_3(W))$.

\[
F_4(W) = \text{lub}_{\sigma \in \omega}(T_4 \uparrow n)
\]
(by the previous proof)

\[
= \text{lub}_{\sigma \in \omega}(\text{Min}(T_3 \uparrow n))
\]
(by definition)

\[
= \text{Min}
\left( \bigcup_{n \in \omega} \text{Min}(T_3 \uparrow n) \right)
\]

\[
\cup \bigvee \left\{ I \in \{ \text{Min}(T_3 \uparrow n) \} \bigg| \text{Min}
\left( \bigcup_{n \in \omega} \text{Min}(T_3 \uparrow n) \right) \subseteq I \right\}
\]

\[
= \text{Min}
\left( \bigcup_{n \in \omega} T_3 \uparrow n \right)
\]

\[
\cup \bigvee \left\{ \text{Min}(T_3 \uparrow n) \big| \text{Min}
\left( \bigcup_{n \in \omega} \text{Min}(T_3 \uparrow n) \right) \subseteq \text{Min}(T_3 \uparrow n) \right\}
\]

\[
= \text{Min}
\left( \bigcup_{n \in \omega} T_3 \uparrow n \right).
\]

It is worth noting that also the properties corresponding to those in Corollary 4.29 hold, as well as the ordering corresponding to the one in Theorem 4.30. Figure 4 shows the relations between the various fixpoints (Proposition 5.17) and the various transformations (Proposition 5.3).

**Theorem 5.18** (Falaschi et al., 1989). For every program $W$, $M_3(W) = \text{lfp}(T_3(W)) = F_3(W)$.

**Theorem 5.19** (Equivalence of Model-Theoretic and Fixpoint Semantics). For $i = 1, 2, 3, 4$, $F_i(W) = M_i(W)$.

**Proof.** $F_3(W) = M_3(W)$ by Theorem 5.18. The rest follows from Propositions 4.28 and 5.17.
Let us give a simple example, which shows how the various models are different.

**Example 5.20.** Let $W$ be the program

\[
\{p(X), p(a)\}
\]

in the language defined by $C = \{a \setminus 0, b \setminus 0\}$ and $P = \{p \setminus 1\}$,

\[
M_1(W) = \{p(a), p(b)\}
\]

\[
M_2(W) = \{p(X), p(a), p(b)\}
\]

\[
M_3(W) = \{p(X), p(a)\}
\]

\[
M_4(W) = \{p(X)\}.
\]

6. **Program Equivalences**

In this section we discuss the previously introduced four models as abstraction operators; namely, as operators inducing equivalence relations on programs.

**Definition 6.1 (Equivalence Relations).** Let $W, W'$ be programs. For $i = 1, 2, 3, 4$, we define

\[
W \equiv_i W' \text{ iff } M_i(W) = M_i(W').
\]
It is easy to see that the $\equiv_i$'s are equivalence relations. They are ordered as shown by the following proposition.

**Proposition 6.2.** $\equiv_3 \subseteq \equiv_2 \subseteq \equiv_1$; i.e., $\equiv_3$ is finer than $\equiv_2$, and $\equiv_2$ is finer than $\equiv_1$. Moreover $\equiv_4 \equiv \equiv_2$.

**Proof.** (\(\equiv_3 \subseteq \equiv_2\)) If \(W \equiv_3 W'\) then \(M_3(W) = M_3(W')\). Then \([M_3(W)] = \text{Up}(M_3(W'))\) holds. By Proposition 4.28, \(M_2(W) = M_2(W')\); that is, \(W \equiv_2 W'\).

(\(\equiv_2 \subseteq \equiv_1\)) If \(W \equiv_2 W'\) then \(M_2(W) = M_2(W')\). Then \([M_2(W)] = [M_2(W')]\) holds. By Corollary 4.29, \(M_1(W) = M_1(W')\); that is, \(W \equiv_1 W'\).

(\(\equiv_4 \subseteq \equiv_2\)) If \(W \equiv_4 W'\) then \(M_4(W) = M_4(W')\). Then \([M_4(W)] = \text{Up}(M_4(W'))\) holds. By Corollary 4.29, \(M_2(W) = M_2(W')\); that is, \(W \equiv_2 W'\).

(\(\equiv_3 \subseteq \equiv_4\)) If \(W \equiv_3 W'\), then \(M_2(W) = M_2(W')\). Then \(\text{Min}(M_2(W)) = \text{Min}(M_2(W'))\) holds. By Corollary 4.29, \(M_4(W) = M_4(W')\); that is, \(W \equiv_4 W'\).

7. **Relation between the Declarative and the Operational Semantics**

In this section we give a soundness and completeness theorem, which fully characterizes the correspondence between the model-theoretic and the operational semantics.

**Theorem 7.1 (Soundness and Completeness).** For \(i = 1, 2, 3\), \(M_i(W) = O_i(W)\).

**Proof.** (\(i = 1\)) \(O_1(W)\) is the standard success set of \(W\), and \(M_1(W)\) is the standard least Herbrand model. Then the results follows from the completeness theorem for ground atoms (Lloyd, 1987; Apt, 1990).

(\(i = 3\)) \(M_3(W)\) is the least S-model in Falaschi et al. (1988, 1989). The soundness Theorem 7.1 in Falaschi et al. (1989) states that

If \(\leftarrow A \rightarrow\sigma\varnothing\), then \(\exists A' \in M_3(W)\) such that \(\varnothing = \text{mgu}(A, A')\), \(\sigma\).

(1)

where \(\sigma|_B\) denotes the substitution \(\sigma\) restricted to the variables occurring in \(B\). The completeness Theorem 7.7 in Falaschi et al. (1989) states that

If \(A' \in M_3(W), A \in B,\) and \(\varnothing = \text{mgu}(A, A')\), then \(\leftarrow A \rightarrow\sigma\varnothing\).

(2)

\((O_3(W) \subseteq M_3(W))\) Let \(A \in O_3(W)\). By definition, \(A = p(X_1, \ldots, X_n)\), where \(\leftarrow p(X_1, \ldots, X_n) \rightarrow\sigma\varnothing\). By (1), there exists \(A' \in M_3(W)\) such that
\[ g = \text{mgu}(p(X_1, ..., X_n), A') \] for \( p(X_1, ..., X_n) \leq A' \). Let \( g = \{ X_i \setminus t_i, ..., X_n \setminus t_n \} \). Since \( p(X_1, ..., X_n) \leq A' \), then \( A' = p(t_1, ..., t_n) = p(X_1, ..., X_n) g = A \).

\((M_3(W) \subseteq O_3(W)) \) Let \( A = p(t_1, ..., t_n) \in M_3(W) \). Let \( g = \text{mgu}(p(X_1, ..., X_n), p(t_1, ..., t_n)) \), \( p(X_1, ..., X_n) = \{ X_1 \setminus t_1, ..., X_n \setminus t_n \} \). By (2), \( \leftarrow p(X_1, ..., X_n) \vdash g \vdash \square \). Then \( A = p(X_1, ..., X_n) g \in O_3(W) \).

\((i = 2) \) By Proposition 4.28, it is sufficient to show that \( [M_3(W)] = O_2(W) \), or equivalently (by the case \( i = 3 \) above), that \( [O_3(W)] = O_2(W) \).

\(([O_3(W)] \subseteq O_2(W)) \) Let \( A' \in [O_3(W)] \). Then there exists \( A \in O_3(W) \) such that \( A \leq A' \). By the Strong Soundness theorem, \( A \) is a logical consequence of \( W \). Then \( A' \) is also a logical consequence of \( W \). Therefore, by Clark's Strong Completeness theorem, \( \leftarrow A' \vdash \square \); that is, \( A' \in O_2(W) \).

\((O_2(W) \subseteq [O_3(W)] \) Let \( A = p(t_1, ..., t_n) \in O_2(W) \). Then \( A \) is a logical consequence of \( W \). Let \( X_1, ..., X_n \in V \). By Clark's Strong Completeness theorem, \( \leftarrow p(X_1, ..., X_n) \vdash g \vdash \square \) for some substitution \( g \) such that \( p(X_1, ..., X_n) g \leq A \). Therefore, \( A \in [O_3(W)] \).

The following corollary shows that the equivalences induced by the model-theoretic and fixpoint semantics exactly correspond to those induced by the operational semantics.

**Corollary 7.2.** For \( i = 1, 2, 3 \), \( \approx_i = \equiv_i \).

**Proof.** This follows immediately from Theorem 7.1.

8. CONCLUSION

In this paper we have defined a notion of truth on Herbrand interpretations extended with variables and a complete partial order, which allow us to capture, by means of suitable models, various operational properties. Our construction has several nice properties:

- The Herbrand models are models. There exists the least model \( \langle M_1(W) \rangle \) which is the same as the least Herbrand model and is equivalent to the ground success set operational semantics.

- The S-models defined in Falaschi et al. (1989) are models. The least S-model \( \langle M_3(W) \rangle \), the S-semantics in Falaschi et al. (1989)), is the same as the derivable atoms semantics in Gaifman and Shapiro (1989b) and is equivalent to the computed answer substitution operational semantics.
The C-models defined in Falaschi et al. (1989) are models. The least C-model \((M_2(W))\), the C-semantics in Falaschi et al. (1989), is the same as Clark's semantics (Clark, 1979) and is equivalent to the non-ground success set operational semantics. \(M_2(W)\) is in one to one correspondence with another model \((M_4(W))\), which is the atomic consequences semantics in Gaifman and Shapiro (1989a).

Each of these four interesting models can be obtained as the least fixpoint of a suitable transformation on the complete lattice of interpretations.

\(M_3(W)\) is the model which has the richest information content. In fact, the other models can be obtained by applying suitable abstraction operators, and not vice versa. This is also shown by the fact that the program equivalence relation based on \(M_3(W)\) is finer than those based on the other models. \(M_3(W)\), which was already noted to define the correct semantics for definite clauses viewed as a programming language (Falaschi et al., 1988), results to be a non-minimal model. This shows that in general it could be true that minimal models are adequate from a logical point of view, but some richer models are needed to cope with the typical programming language features, i.e., observable behaviours.

The usefulness of \(M_3(W)\) has already been shown by several projects related to the semantics, the analysis, and the transformation of logic programs. These include:

- Characterization of the non-ground finite failure set (Levi et al., 1990).
Other promising areas of current research include:

- The fixpoint semantics of perpetual logic processes.
- The extension of the theory to other observable properties (for example, finite failures) and to language features closer to those of sequential Prolog (for example, with clause ordering and sequences of computed answer substitutions as observable properties).

REFERENCES


