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).

1. INTRODUCTION

The least Herbrand model semantics was originally proposed (van 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 $\vartheta: V \to T$ such that the set $D(\vartheta) = \{X \mid \vartheta(X) \neq X\}$ (*domain of* ϑ) is finite. ε denotes the empty substitution. The composition $\vartheta\sigma$ of the substitutions ϑ and σ is defined as the functional composition. The pre-ordering $\leqslant$ on substitutions is such that $\vartheta \leqslant \sigma$ iff there exists ϑ' such that $\vartheta\vartheta' = \sigma$.

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

A *definite clause* in the language L is a formula $H \leftarrow B_1, ..., B_n$ $(n \geqslant 0)$, 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 $B_1, ..., B_n$ is the *body*. Given a clause $H \leftarrow B_1, ..., B_n$, a substitution ϑ is *grounding* if $H\vartheta, B_1\vartheta, ..., B_n\vartheta$ 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 $W = \{C_1, ..., C_n\}$ in L. A *goal statement* is a formula $\leftarrow A_1, ..., A_m$, where each A_i is an atom in L.

$G \overset{\vartheta}{\longmapsto} \square$ denotes the SLD-refutation of a goal G with computed answer substitution ϑ.

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

$$O_1(W) = \{A \mid A \text{ is ground and } \leftarrow A \overset{\varepsilon}{\longmapsto} \Box\}.$$

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 $P = \{p\backslash 1, q\backslash 1\}$:

$$\begin{aligned} W_1 = \{ & p(a). \\ & q(X).\} \\ W_2 = \{ & p(a). \\ & q(a).\}. \end{aligned}$$

$W_1 \approx_1 W_2$ 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:

$$O_2(W) = \{A \mid \leftarrow A \overset{\varepsilon}{\longmapsto} \Box\}$$

$O_2(W)$ 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:

$$\begin{aligned} W_1 = \{&p(a).\\ &q(X).\}\\ W_2 = \{&p(a).\\ &q(X).\\ &q(a).\}. \end{aligned}$$

$W_1 \approx_2 W_2$ 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

$$\begin{aligned} O_3(W) = \{A \mid\ &\exists p \in P,\\ &\exists X_1, ..., X_n \text{ distinct variables in } V, \exists \vartheta,\\ &\leftarrow p(X_1, ..., X_n) \stackrel{\vartheta}{\longmapsto} \Box,\\ &A = p(X_1, ..., X_n)\, \vartheta.\} \end{aligned}$$

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 $\mathbf{B}$ is the quotient set of all the atoms with respect to $\approx$. The ordering induced by $\leqslant$ on $\mathbf{B}$ will still be denoted by $\leqslant$. For the sake of simplicity, we will represent the equivalence class of an atom A by A itself.

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

It is worth noting that $\mathbf{B}$ (and therefore the set of interpretations $\mathscr{I}$) 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	$\lceil I \rceil = \{A \in \mathbf{B} \mid \exists A' \in I, A' \leqslant A\}$,
ground atoms	$\lfloor I \rfloor = \{A \in I \mid A \text{ is ground}\}$
minimal elements	$\mathrm{Min}(I) = \{A \in I \mid \forall A' \in I \text{ if } A' \leqslant A \text{ then } A = A'\}$.

Let us introduce the notation $[I]$ as a shorthand for $\lfloor \lceil I \rceil \rfloor$. 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 $[\{A\}] \subseteq [I]$. Note also that if $A \in \lceil I \rceil$ 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, $\lceil I \rceil = \lceil \mathrm{Min}(I) \rceil$ and $\mathrm{Min}(\lceil I \rceil) = \mathrm{Min}(I)$ 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 $A' \leqslant A$ up to variable renaming. ∎

LEMMA 3.9. *Let A be any atom and I be any interpretation. A is true in $\lceil I \rceil$ iff A is true in I.*

Proof. Straightforward, since $[\lceil I \rceil] = [I]$. ∎

LEMMA 3.10. *For any interpretation I, $\lceil I \rceil$ 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 $\mathrm{Min}(I)$ iff A is true in I.*

Proof. Immediate (by Lemma 3.8), since $\ulcorner \mathrm{Min}(I) \urcorner = \ulcorner I \urcorner$. ∎

LEMMA 3.12. *For any interpretation I,* 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 $C = \{a\backslash 0, f\backslash 1\}$ and $P = \{p\backslash 1, q\backslash 1\}$:

$$\begin{aligned} W = \{ & p(f(a)). \\ & p(X). \\ & q(a). \quad \} \end{aligned}$$

$$\begin{aligned} O_1(W) &= \{q(a), \quad p(a), \quad p(f(a)), \quad p(f(f(a))), \ldots\} \\ O_2(W) &= \{q(a), p(X), p(a), p(f(X)), p(f(a)), p(f(f(X))), p(f(f(a))), \ldots\} \\ O_3(W) &= \{q(a), p(X), \quad p(f(a)) \quad \}. \end{aligned}$$

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 interpretation $\{p(X)\}$, 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:

- $I_1 \leqslant I_2$ iff $\forall A_1 \in I_1, \exists A_2 \in I_2$ such that $A_2 \leqslant A_1$.
- $I_1 \sqsubseteq I_2$ iff $(I_1 \leqslant I_2)$ and $(I_2 \leqslant I_1$ implies $I_1 \subseteq I_2)$.

The intuitive meaning of the above defined relations is the following: $I_1 \leqslant I_2$ means that every atom verified by I_1 is also verified by I_2 (I_2 contains more *positive information*). Note that $\leqslant$, 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 ($\leqslant$ Is a Preorder). *The relation* $\leqslant$ *of Definition* 4.1 *is a preorder.*

LEMMA 4.3. (a) *If* $I_1 \subseteq I_2$ *then* $I_1 \leqslant I_2$.

(b) $I_1 \leqslant I_2$ *iff* $\lceil I_1 \rceil \subseteq \lceil I_2 \rceil$.

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, $I \leqslant I$ (by Lemma 4.2) and $I \subseteq I$.

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

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

PROPOSITION 4.5. *If* $I_1 \subseteq I_2$, *then* $I_1 \sqsubseteq I_2$.

Proof. Immediate, since, by Lemma 4.3(a), $I_1 \subseteq I_2$ implies $I_1 \leqslant I_2$. ∎

EXAMPLE 4.6. Let the base **B** be the set $\{p(X), p(a), p(b)\}$. The set $\mathscr{I}$ of interpretations is ordered as shown in Fig. 1, where a directed arc from I_1 to I_2 denotes $I_1 \sqsubseteq I_2$.

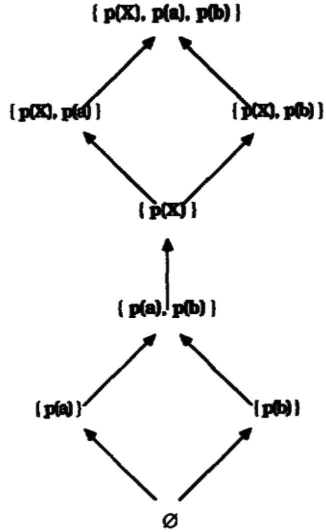

FIG. 1. A partially ordered set of interpretations.

Now we discuss the properties of the relation $\sqsubseteq$ on interpretations. We show that the set of interpretations under the relation $\sqsubseteq$ is a complete lattice.

DEFINITION 4.7. Let Γ be a set of interpretations. We introduce the following notations:

- $\nabla\Gamma = \bigcup_{I \in \Gamma} I$
- $\mathrm{Min}(\Gamma) = \mathrm{Min}(\nabla\Gamma)$
- $\bigsqcup \Gamma = \mathrm{Min}(\Gamma) \cup \nabla\{I \in \Gamma \mid \mathrm{Min}(\Gamma) \subseteq I\}$.

Note that $\mathrm{Min}(\Gamma) = \mathrm{Min}(\bigsqcup \Gamma)$.

PROPOSITION 4.8. *For any set Γ of interpretations there exists the least upper bound of Γ,* $\mathrm{lub}(\Gamma)$, *and* $\mathrm{lub}(\Gamma) = \bigsqcup \Gamma$ *holds.*

Proof. 1. ($\bigsqcup \Gamma$ is an upper bound of Γ) If I is an element of Γ, then $\lceil \mathrm{Min}(I) \rceil \subseteq \lceil \mathrm{Min}(\Gamma) \rceil = \lceil \mathrm{Min}(\bigsqcup \Gamma) \rceil$. Therefore, by Lemma 3.8, $\lceil I \rceil \subseteq \lceil \bigsqcup \Gamma \rceil$ holds. Then, by Lemma 4.3(b), $I \leqslant \bigsqcup \Gamma$. Moreover, if $\bigsqcup \Gamma \leqslant I$, then, by Lemma 4.3(b), $\lceil \bigsqcup \Gamma \rceil \subseteq \lceil I \rceil$. Then $\mathrm{Min}(\Gamma) = \mathrm{Min}(\lceil \bigsqcup \Gamma \rceil) \subseteq I$. Therefore, by Definition of $\bigsqcup \Gamma$, $I \subseteq \bigsqcup \Gamma$.

2. ($\bigsqcup \Gamma$ is the least upper bound of Γ) Let H be an upper bound of Γ. Since, for every $I \in \Gamma$ (by Lemma 4.3(b)) $\lceil I \rceil \subseteq \lceil H \rceil$, then $\lceil \bigsqcup \Gamma \rceil \subseteq \lceil H \rceil$;

i.e., $\bigsqcup \Gamma \leqslant H$. Assume now $H \leqslant \bigsqcup \Gamma$; i.e., $\lceil H \rceil \subseteq \lceil \bigsqcup \Gamma \rceil$. Then $\lceil H \rceil = \lceil \bigsqcup \Gamma \rceil$ and therefore $\mathrm{Min}(\Gamma) = \mathrm{Min}(\bigsqcup \Gamma) \subseteq H$. Moreover, for any $I \in \Gamma$ such that $\mathrm{Min}(\Gamma) \subseteq I$, $\lceil H \rceil \subseteq \lceil I \rceil$ and therefore (since $I \sqsubseteq H$) $I \subseteq H$. Then $\nabla\{I \in \Gamma \mid \mathrm{Min}(\Gamma) \subseteq I\} \subseteq H$ and therefore $\bigsqcup \Gamma \subseteq H$ holds. ∎

THEOREM 4.9. *The set of all the interpretations $\mathscr{I}$ with the ordering $\sqsubseteq$ is a complete lattice.* **B** *is the top element and $\varnothing$ is the bottom element.*

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

$$\mathrm{glb}(\Gamma) = \mathrm{lub}(\{I \in \mathscr{I} \mid \forall I' \in \Gamma, I \sqsubseteq I'\}). \quad ∎$$

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* **M** *be a set of models of a program W. Then* glb(**M**) *is a model of W.*

Proof. Let $A \leftarrow B_1, ..., B_n$ be a clause of W. Consider a substitution ϑ. Assume that $B_1\vartheta, ..., B_n\vartheta$ are true in glb(**M**). Then $[\{B_1\vartheta, ..., B_n\vartheta\}] \subseteq \lceil \mathrm{glb}(\mathbf{M}) \rceil$. By definition of glb, for any $I \in \mathbf{M}$, $\mathrm{glb}(\mathbf{M}) \sqsubseteq I$ holds. Therefore $\lceil \mathrm{glb}(\mathbf{M}) \rceil \subseteq \lceil I \rceil$ and then $[\{B_1\vartheta, ..., B_n\vartheta\}] \subseteq \lceil I \rceil$. Since $A \leftarrow B_1, ..., B_n$ is true in I, $[\{A\vartheta\}] \subseteq \lceil I \rceil$, by definition. This implies, by Proposition 4.5, $[\{A\vartheta\}] \sqsubseteq \lceil I \rceil$. Hence, $[\{A\vartheta\}]$ is a lower bound of **M**, and therefore $[\{A\vartheta\}] \sqsubseteq \mathrm{glb}(\mathbf{M})$. By definition, $\lceil [\{A\vartheta\}] \rceil \subseteq \lceil \mathrm{glb}(\mathbf{M}) \rceil$ and therefore $[\{A\vartheta\}] = \lceil [\{A\vartheta\}] \rceil \subseteq \lceil \mathrm{glb}(\mathbf{M}) \rceil$. ∎

COROLLARY 4.11. *The set of all the models of a program W with the ordering $\sqsubseteq$ 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., $M_1(W) = \mathrm{glb}(\{I \in \mathscr{I} \mid I$ 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 $I' \sqsubseteq I$.*

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

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

Proof. Let I, I' range on the set of models and on the set of Herbrand models respectively. Define Φ and Ψ 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. ∎

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. ∎

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 $\mathbf{B}$ such that $I = \lceil I \rceil$.

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

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

- A (possibly non-ground) atom A is S-true in I iff $\exists A' \in \lceil I \rceil, A' \leqslant A$,
- A definite clause $A \leftarrow B_1, ..., B_n$ is S-true in I iff for each $B'_1, ..., B'_n \in I$ if $\vartheta = \mathrm{mgu}((B_1, ..., B_n), (B'_1, ..., B'_n))$, then $A\vartheta \in I$.

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 $A \in I = \lceil I \rceil$. Then $\lceil A \rceil \subseteq \lceil I \rceil$, and then $[\{A\}] \subseteq [I]$.

(non-atomic case) Consider a clause $A \leftarrow B_1, ..., B_n$ and a C-interpretation I. For each substitution ϑ, if $B_1\vartheta, ..., B_n\vartheta$ are true in I, i.e., if $([\{B_1\vartheta, ..., B_n\vartheta\}] \subseteq [I])$, we must show that $[\{A\vartheta\}] \subseteq [I]$. Consider a substitution γ such that $A\vartheta\gamma$ is ground. Then $[\{B_1\vartheta\gamma, ..., B_n\vartheta\gamma\}] \subseteq \lceil I \rceil$. Let η be a substitution such that $B_1\vartheta\gamma\eta, ..., B_n\vartheta\gamma\eta$ are ground. Therefore $B_1\vartheta\gamma\eta, ..., B_n\vartheta\gamma\eta \in \lceil I \rceil$. Since I is a C-interpretation, $A\vartheta\gamma = A\vartheta\gamma\eta \in \lceil I \rceil$ for every γ such that $A\vartheta\gamma$ is ground. Therefore $[\{A\vartheta\}] \subseteq \lceil I \rceil$. ∎

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

PROPOSITION 4.19 (Falaschi *et al.*, 1989). *Let $I \in \mathscr{I}$. 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 $\lceil I \rceil$ 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, $M_2(W) = \lceil M_3(W) \rceil$.*

THEOREM 4.24. *Every S-model is a model.*

Proof. By Proposition 4.20, if I is an S-model, then $\lceil I \rceil$ 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 $\mathscr{I}$:

$$\mathscr{M}_H = \{I \in \mathscr{I} \mid I \text{ is a Herbrand model of } W\},$$

$$\mathscr{M}_1 = \{I \in \mathscr{I} \mid I \text{ is a model of } W\},$$

$$\mathscr{M}_2 = \{I \in \mathscr{I} \mid I \text{ is a C-model of } W\},$$

$$\mathscr{M}_3 = \{I \in \mathscr{I} \mid I \text{ is an S-model of } W\}.$$

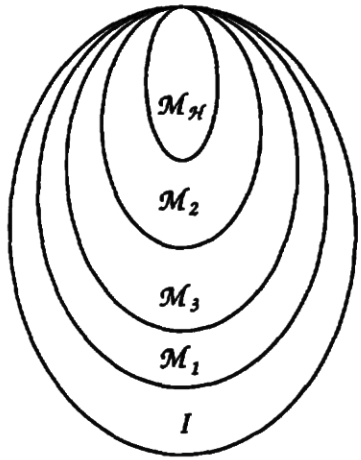

FIG. 2. Models, Herbrand models, S-models, and C-models.

The following facts are true: $\mathcal{M}_H \subseteq \mathcal{M}_2$ (immediate, since if $I \in \mathcal{M}_H$ then $I = \ulcorner I \urcorner$), $\mathcal{M}_2 \subseteq \mathcal{M}_1$ (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:

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

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.

($i = 2, 3$) $\bigcap \mathcal{M}_i \in \mathcal{M}_i$ by Proposition 4.21. Moreover, $\bigcap \mathcal{M}_i$ is a lower bound of $\mathcal{M}_i$, by Proposition 4.5. Therefore $\bigcap \mathcal{M}_i =$ the least element of $\mathcal{M}_i = \mathrm{glb}(\mathcal{M}_i)$. ∎

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*:

- $M_1(W) = [M_3(W)]$
- $M_2(W) = \lceil M_3(W) \rceil$
- $M_4(W) = \mathrm{Min}(M_3(W))$.

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*:

- $M_1(W) = \lfloor M_2(W) \rfloor$
- $M_2(W) = \lceil M_4(W) \rceil$
- $M_4(W) = \mathrm{Min}(M_2(W))$.

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*,

$$M_1(W) \sqsubseteq M_4(W) \sqsubseteq M_3(W) \sqsubseteq M_2(W).$$

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 $M_4(W) =$

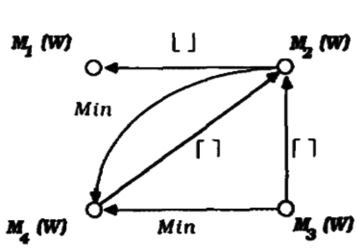

FIG. 3. Models and their mappings.

$\text{Min}(M_3(W)) \subseteq M_3(W)$. 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 $M_2(W)$. 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 $\mathcal{I}$).

1. $$T_1(I) = \{A \in \mathbf{B} \mid \begin{array}{l} A \text{ is ground}, \\ \exists A' \leftarrow B_1, ..., B_n \in W, \\ \exists \vartheta \text{ grounding}, \\ B_1\vartheta, ..., B_n\vartheta \in [I], \\ A = A'\vartheta\} \end{array}$$

2. $$T_2(I) = \{A \in \mathbf{B} \mid \begin{array}{l} \exists A' \leftarrow B_1, ..., B_n \in W, \\ \exists \vartheta, \\ B_1\vartheta, ..., B_n\vartheta \in \lceil I \rceil, \\ A = A'\vartheta\} \end{array}$$

3. $$T_3(I) = \{A \in \mathbf{B} \mid \begin{array}{l} \exists C = A' \leftarrow B_1, ..., B_n \in W, \\ \exists B'_1, ..., B'_n \text{ variants of atoms in } I, \\ \text{with no variables in common} \\ \text{with } C \text{ and with each other}, \\ \exists \vartheta = \text{mgu}((B_1, ..., B_n), (B'_1, ..., B'_n)), \\ A = A'\vartheta\} \end{array}$$

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

LEMMA 5.2.

- $T_1(I) = T_1(\lfloor I \rfloor)$
- $T_2(I) = T_2(\lceil I \rceil)$.

Proof. Immediate.

PROPOSITION 5.3. *The following relations among the various transformations hold*:

- $T_1(I) = \lfloor T_3(I) \rfloor$
- $T_2(I) = \lceil T_3(I) \rceil$
- $T_4(I) = \mathrm{Min}(T_3(I))$.

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 $\langle \mathscr{I}, \sqsubseteq \rangle$.*

Proof. (monotonicity) Let $I \sqsubseteq I'$. Then, $\lceil I \rceil \subseteq \lceil I' \rceil$ and therefore $T_1(I) \subseteq T_1(I')$. By Proposition 4.5 we derive $T_1(I) \sqsubseteq T_1(I')$.

(continuity) $\mathrm{lub}_{I \in \Gamma} T_1(I) \sqsubseteq T_1(\mathrm{lub}(\Gamma))$ follows by monotonicity. Therefore it is sufficient to show that for any chain Γ, $T_1(\mathrm{lub}(\Gamma)) \sqsubseteq \mathrm{lub}_{I \in \Gamma} T_1(I)$ holds. It is easy to see that $T_1(\nabla \Gamma) \subseteq \bigcup_{I \in \Gamma} T_1(I)$. Indeed, $\{\lceil I \rceil \mid I \in \Gamma\}$ is a chain ordered by set inclusion, $T_1(I) = T_1(\lceil I \rceil)$, and T_1 is continuous with respect to set inclusion. Moreover, since $T_1(I) = \lfloor T_1(I) \rfloor$, $\bigcup_{I \in \Gamma} T_1(I)$ is ground. Note that if I is ground $\mathrm{Min}(I) = I$, and the $\mathrm{lub}_{I \in \Gamma} I$, when all the I's are ground, is $\mathrm{Min}(\Gamma) \cup \nabla\{J \in \Gamma \mid \mathrm{Min}(\Gamma) \subseteq J\} = \Gamma$. Thus $\bigcup_{I \in \Gamma} T_1(I) = \mathrm{lub}_{I \in \Gamma} T_1(I)$ holds. Finally, note that $\mathrm{lub}(\Gamma) \subseteq \nabla \Gamma$. Therefore, $T_1(\mathrm{lub}(\Gamma)) \subseteq T_1(\nabla \Gamma) \subseteq \bigcup_{I \in \Gamma} T_1(I) = \mathrm{lub}_{I \in \Gamma} T_1(I)$. By Proposition 4.5, $T_1(\mathrm{lub}(\Gamma)) \sqsubseteq \mathrm{lub}_{I \in \Gamma} T_1(I)$. ▮

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 $\langle \mathscr{I}, \subseteq \rangle$.*

PROPOSITION 5.6 (Monotonicity and Continuity of T_4). *T_4 is monotonic and continuous in the complete lattice $\langle \mathscr{I}, \sqsubseteq \rangle$.*

Proof. (monotonicity) By definition and Lemma 3.8 $T_4(I) = \mathrm{Min}(T_2(I))$. T_2 can be shown to be monotonic in the complete lattice $\langle \mathscr{I}, \sqsubseteq \rangle$ by exactly the same arguments used in Proposition 5.4. Let $I \sqsubseteq I'$. Then $\lceil T_4(I) \rceil = \lceil T_2(I) \rceil \subseteq \lceil T_2(I') \rceil = \lceil T_4(I') \rceil$. Assume now that $\lceil T_4(I') \rceil \subseteq \lceil T_4(I) \rceil$. Then $\lceil \mathrm{Min}(T_2(I')) \rceil \subseteq \lceil \mathrm{Min}(T_2(I)) \rceil$. By Lemma 3.8, $T_2(I') \subseteq T_2(I)$. Hence $T_2(I') = T_2(I)$, and therefore $T_4(I) = \mathrm{Min}(T_2(I)) = \mathrm{Min}(T_2(I')) = T_4(I')$.

(continuity) T_2 is continuous in the complete lattice $\langle \mathscr{I}, \subseteq \rangle$ and $T_2(I) = T_2(\lceil I \rceil)$. Then, for any chain Γ, $T_2(\nabla \Gamma) \subseteq \bigcup_{I \in \Gamma} T_2(I)$, and therefore $\lceil T_4(\mathrm{lub}(\Gamma)) \rceil$ (by Lemma 3.8) $= \lceil T_2(\mathrm{lub}(\Gamma)) \rceil = \lceil T_2(\lceil \mathrm{lub}(\Gamma) \rceil) \rceil = T_2(\nabla \Gamma) \rceil \subseteq \lceil \bigcup_{I \in \Gamma} T_2(I) \rceil$ (since $\mathrm{lub}(\Gamma') = \bigcup_{I \in \Gamma'} I$ when Γ' is a chain of upward closed interpretations) $= \lceil \mathrm{lub}_{I \in \Gamma} T_2(I) \rceil = \lceil \mathrm{lub}_{I \in \Gamma} T_4(I) \rceil$. Assume now that $\lceil \mathrm{lub}_{I \in \Gamma} T_4(I) \rceil \subseteq \lceil T_4(\mathrm{lub}(\Gamma)) \rceil$. Then $\mathrm{Min}(T_4(\mathrm{lub}(\Gamma))) = \mathrm{Min}(\mathrm{lub}_{I \in \Gamma} T_4(I))$. Since $\mathrm{Min}(T_4(\mathrm{lub}(\Gamma))) = T_4(\mathrm{lub}(\Gamma))$, $T_4(\mathrm{lub}(\Gamma)) \subseteq \mathrm{lub}_{I \in \Gamma} T_4(I)$. By Proposition 4.5 $T_4(\mathrm{lub}(\Gamma)) \sqsubseteq \mathrm{lub}_{I \in \Gamma} T_4(I)$. ∎

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

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

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

THEOREM 5.8. *For $i = 1, 2, 3, 4$, $T_i \uparrow \omega$ is the least fixpoint of T_i in the complete lattice $\langle \mathscr{I}, \sqsubseteq \rangle$.*

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 $\langle \mathcal{I}, \subseteq \rangle$ in the case of T_2 and T_3.

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

EXAMPLE 5.9. Consider the program W in the language defined by $C = \{a\backslash 0\}$ and $P = \{p\backslash 1\}$:

$$W = \{p(X).$$
$$p(X) \leftarrow p(X).\}.$$

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

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

EXAMPLE 5.10. Let W be the program

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

in the language L defined by $C = \{0\backslash 0, s\backslash 1\}$ and $P = \{p\backslash 2\}$,

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

Then $\mathrm{lub}_n(T_3 \uparrow n)$ 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

$$\{p(0, X).$$
$$p(s(Y), X) \leftarrow p(Y, X).\}$$

in the language L defined by $C = \{0\backslash 0, s\backslash 1\}$ and $P = \{p\backslash 2\}$,

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

$$\dots$$

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

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

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

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 $T_3(I) \subseteq I$.*

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 $I = \lfloor I \rfloor$ 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 $\mathrm{Min}(\ulcorner J \urcorner)$ for some J $(T_4(I) = \mathrm{Min}(T_2(I)) = \mathrm{Min}(\ulcorner T_2(I) \urcorner) = I)$. Thus $T_4(\mathrm{Min}(\ulcorner J \urcorner)) = \mathrm{Min}(\ulcorner J \urcorner)$.

By definition of T_4, $\mathrm{Min}(T_2(\mathrm{Min}(\ulcorner J \urcorner))) = \mathrm{Min}(\ulcorner J \urcorner)$.

Since $T_2(\mathrm{Min}(I)) = T_2(I)$ (see Lemma 3.8 and definition of T_2), $\mathrm{Min}(T_2(\ulcorner J \urcorner)) = \mathrm{Min}(\ulcorner J \urcorner)$.

Since both the arguments of Min are closed under the Up operation, we can deduce that $T_2(\ulcorner J \urcorner) = \ulcorner J \urcorner$; i.e., $\ulcorner J \urcorner$ is a fixpoint of T_2, and thus it is a model. Finally, by Lemma 3.12 also $\mathrm{Min}(\ulcorner J \urcorner) = I$ 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. $\mathrm{Min}(T_3(\mathrm{Min}(I))) = \mathrm{Min}(T_3(I))$.

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

$(\supseteq)$ $A \in \mathrm{Min}(T_3(I))$ implies $\exists A' \leftarrow B_1, ..., B_n$, where $B'_1, ..., B'_n \in I$, $\vartheta = \mathrm{mgu}((B_1, ..., B_n), (B'_1, ..., B'_n))$ and $A = A'\vartheta$.

There exist $B''_1, ..., B''_n \in \mathrm{Min}(I)$ renamed apart from $B_1, ..., B_n$ and from each other, where $(B''_1, ..., B''_n) \leqslant (B'_1, ..., B'_n)$.

Thus there exists $\vartheta'' = \mathrm{mgu}((B''_1, ..., B''_n), (B_1, ..., B_n))$ and $\vartheta''_{|B_1, ..., B_n} \leqslant \vartheta_{|B_1, ..., B_n}$.

Hence $A'' = A'\vartheta'' \leqslant A \in \mathrm{Min}(T_3(I))$. But $A'' \in T_3(\mathrm{Min}(I)) \subseteq T_3(I)$ implies $A'' = A'$. ∎

PROPOSITION 5.17. *Let W be a program*

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

(b) $F_2(W) = \lceil F_3(W) \rceil$

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

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)

$$\begin{aligned} T_1 \uparrow k+1 &= T_1(T_1 \uparrow k) && \text{(by the inductive hypothesis)} \\ &= T_1([T_3 \uparrow k]) && \text{(by Proposition 5.3)} \\ &= T_1(T_3 \uparrow k) && \text{(by Lemma 5.2)} \\ &= [T_3(T_3 \uparrow k)] \\ &= [T_3 \uparrow k+1]. \end{aligned}$$

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

$$\begin{aligned} F_1(W) &= \mathrm{lub}_{n \in \omega}(T_1 \uparrow n) && (\text{since } T_1 \uparrow n = \lfloor T_1 \uparrow n \rfloor) \\ &= \bigcup_{n \in \omega} T_1 \uparrow n && \text{(by the previous proof)} \\ &= \bigcup_{n \in \omega} [T_3 \uparrow n] && \left(\text{since } \bigcup_{I \in \Gamma} [I] = [\nabla \Gamma]\right) \\ &= \left[\bigcup_{n \in \omega} T_3 \uparrow n\right] \\ &= [F_3(W)]. \end{aligned}$$

(b) The proof is similar to the one of case (a).

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

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

(n = k + 1)

$$
\begin{aligned}
T_4 \uparrow k+1 &= T_4(T_4 \uparrow k) && \text{(by the inductive hypothesis)}\\
&= T_4(\mathrm{Min}(T_3 \uparrow k)) && \text{(by Proposition 5.3)}\\
&= \mathrm{Min}(T_3(\mathrm{Min}(T_3 \uparrow k))) && \text{(by Lemma 5.16)}\\
&= \mathrm{Min}(T_3 \uparrow k+1)).
\end{aligned}
$$

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

$$
\begin{aligned}
F_4(W) &= \mathrm{lub}_{n\in\omega}(T_4 \uparrow n) \qquad \text{(by the previous proof)}\\
&= \mathrm{lub}_{n\in\omega}(\mathrm{Min}(T_3 \uparrow n)) \qquad \text{(by definition)}\\
&= \mathrm{Min}\left(\bigcup_{n\in\omega} \mathrm{Min}(T_3 \uparrow n)\right)\\
&\cup \nabla \left\{ I \in \{\mathrm{Min}(T_3 \uparrow n)\} \,\middle|\, \mathrm{Min}\left(\bigcup_{n\in\omega} \mathrm{Min}(T_3 \uparrow n)\right) \subseteq I \right\}\\
&= \mathrm{Min}\left(\bigcup_{n\in\omega} T_3 \uparrow n\right)\\
&\cup \nabla \left\{ \mathrm{Min}(T_3 \uparrow n) \,\middle|\, \mathrm{Min}\left(\bigcup_{n\in\omega} \mathrm{Min}(T_3 \uparrow n)\right) \subseteq \mathrm{Min}(T_3 \uparrow n) \right\}\\
&= \mathrm{Min}\left(\bigcup_{n\in\omega} T_3 \uparrow n\right). \quad \blacksquare
\end{aligned}
$$

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, $M_3(W) = \mathrm{lfp}(T_3(W)) = F_3(W)$.*

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. $\blacksquare$

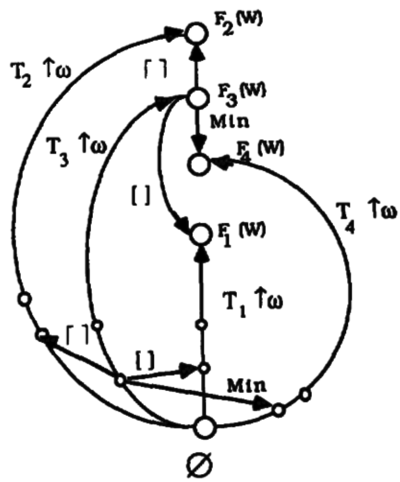

FIG. 4. Fixpoints and transformations.

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 $C = \{a\backslash 0, b\backslash 0\}$ and $P = \{p\backslash 1\}$,

$$M_1(W) = \{p(a), p(b)\}$$
$$M_2(W) = \{p(X), p(a), p(b)\}$$
$$M_3(W) = \{p(X), p(a)\}$$
$$M_4(W) = \{p(X)\}.$$

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 $\equiv_4 = \equiv_2$.*

Proof. ($\equiv_3 \subseteq \equiv_2$) If $W \equiv_3 W'$ then $M_3(W) = M_3(W')$. Then $\ulcorner M_3(W)) = \mathrm{Up}(M_3(W')\urcorner$ 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 $\lfloor M_2(W))\rfloor = \lfloor M_2(W')\rfloor$ 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 $\ulcorner M_4(W) = \mathrm{Up}(M_4(W')\urcorner$ holds. By Corollary 4.29, $M_2(W) = M_2(W')$; that is, $W \equiv_2 W'$.

($\equiv_2 \subseteq \equiv_4$) If $W \equiv_2 W'$, then $M_2(W) = M_2(W')$. Then $\mathrm{Min}(M_2(W)) = \mathrm{Min}(M_2(W'))$ 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

$$\text{If } \leftarrow A \overset{\vartheta}{\longmapsto} \square, \text{ then } \exists A' \in M_3(W) \text{ such that } \vartheta = \mathrm{mgu}(A, A')_{|A}, \tag{1}$$

where $\sigma_{|\beta}$ denotes the substitution σ restricted to the variables occurring in B. The completeness Theorem 7.7 in Falaschi *et al.* (1989) states that

$$\text{If } A' \in M_3(W),\ A \in \mathbf{B}, \text{ and } \vartheta = \mathrm{mgu}(A, A')_{|A}, \text{ then } \leftarrow A \overset{\vartheta}{\longmapsto} \square. \tag{2}$$

($O_3(W) \subseteq M_3(W)$) Let $A \in O_3(W)$. By definition, $A = p(X_1, \ldots, X_n)\,\vartheta$, where $\leftarrow p(X_1, \ldots, X_n) \overset{\vartheta}{\longmapsto} \square$. By (1), there exists $A' \in M_3(W)$ such that

$\vartheta = \mathrm{mgu}(p(X_1, ..., X_n), A')_{|p(X_1, ..., X_n)}$. Let $\vartheta = \{X_1 \backslash t_1, ..., X_n \backslash t_n\}$. Since $p(X_1, ..., X_n) \leqslant A'$, then $A' = p(t_1, ..., t_n) = p(X_1, ..., X_n)\,\vartheta = A$.

($M_3(W) \subseteq O_3(W)$) Let $A = p(t_1, ..., t_n) \in M_3(W)$. Let $\vartheta = \mathrm{mgu}(p(X_1, ..., X_n), p(t_1, ..., t_n))_{|p(X_1, ..., X_n)} = \{X_1 \backslash t_1, ..., X_n \backslash t_n\}$. By (2), $\leftarrow p(X_1, ..., X_n) \overset{\vartheta}{\longmapsto} \Box$. Then $A = p(X_1, ..., X_n)\,\vartheta \in O_3(W)$.

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

($\ulcorner O_3(W) \urcorner \subseteq O_2(W)$) Let $A' \in \ulcorner O_3(W) \urcorner$. Then there exists $A \in O_3(W)$ such that $A \leqslant A'$. 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, $\leftarrow A' \overset{\varepsilon}{\longmapsto} \Box$; that is, $A' \in O_2(W)$.

($O_2(W) \subseteq \ulcorner O_3(W) \urcorner$) 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, $\leftarrow p(X_1, ..., X_n) \overset{\vartheta}{\longmapsto} \Box$ for some substitution ϑ such that $p(X_1, ... X_n)\,\vartheta \leqslant A$. Therefore, $A \in \ulcorner O_3(W) \urcorner$. ∎

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 ($M_1(W)$) 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 ($M_3(W)$, 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 then 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:

• Semantics of concurrent and distributed logic languages (Levi and Palamidessi, 1987; Levi, 1988; de Boer *et al.*, 1989a, 1989b; Brogi and Gorrieri, 1989; Murakami, 1990; Falaschi, Gabbrielli, Levi and Murakami, 1990; Gabbrielli and Levi, 1990).

• Semantics of partial computations (Falaschi and Levi, 1990).

• Abstract interpretation (Barbuti, Giacobazzi, and Levi, 1993; Barbuti and Giacobazzi, 1992; Codish *et al.*, 1990; Giacobazzi and Ricci, 1990; Kemp and Ringwood, 1990).

• Correctness of program transformation techniques (Levi, 1988; Levi and Mancarella, 1988; Bossi and Cocco, 1990).

• Semantics of constraint logic programming (Gabbrielli and Levi, 1991).

• Characterization of the non-ground finite failure set (Levi *et al.*, 1990).

• Semantics of programs with negation (Turi, 1991; Di Pierro *et al.*, 1991).

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

APT, K. R. (1990), Introduction to logic programming, in "Handbook of Theoretical Computer Science, Vol. B, Formal Models and Semantics" (J. Van Leeuwen, Ed.), Elsevier, Amsterdam, and MIT Press, Cambridge, MA.

BARBUTI, R., AND GIACOBAZZI, R. (1992), "A Bottom-up Polymorphic Type Inference in Logic Programming," Science of Computer Programming 19.

BARBUTI, R., GIACOBAZZI, R., AND LEVI, G. (1993), "A General Framework for Semantics-Based Bottom-up Abstract Interpretation of Logic Programs," ACM Transactions on Programming Languages and Systems.

DE BOER, F. S., KOK, J., PALAMIDESSI, C., AND RUTTEN, J. (1989), Semantic models for a version of Parlog, *in* "Proceedings, Sixth International Conference on Logic Programming" (G. Levi and M. Martelli, Eds.), pp. 621–636, MIT Press, Cambridge, MA.

DE BOER, F. S., KOK, J., PALAMIDESSI, C., AND RUTTEN, J. (1989), Control flow versus logic: A denotational and a declarative model for guarded Horn clauses, *in* "Proceedings, Mathematical Foundations of Computer Science, 1989" (A. Kreczmar and G. Mirkowska, Eds.), pp. 165–176, Lecture Notes in Computer Science, Vol. 379, Springer-Verlag, Berlin/New York.

BOSSI, A., AND COCCO, N. (1990), "Basic Transformation Operations for Logic Programs Which Preserve Computed Answer Substitutions," Technical Report 16, Dipartimento di Matematica Pura ed Applicata, Università di Padova. To appear in the special issue on partial deduction of *J. Logic Progr.*

BROGI, A., AND GORRIERI, R. (1989), Model-theoretic, fixpoint and operational semantics for a distributed logic language, *in* "Proceedings, Sixth International Conference on Logic Programming" (G. Levi and M. Martelli, Eds.), pp. 637–652, MIT Press, Cambridge, MA.

CLARK, K. L. (1979), "Predicate Logic as a Computational Formalism," Research Report DOC 79/59, Department of Computing, Imperial College, London.

CODISH, M., DAMS, D., AND YARDENI, E. (1990), "Bottom-up Abstract Interpretation of Logic Programs," to appear in *Theoret. Comput. Sci.*

DI PIERRO, A., MARTELLI, M., AND PALAMIDESSI, C. (1991), Negation as instantiation: A new rule for the treatment of negation in logic programming, *in* "Proceedings, Eight International Conference on Logic Programming" (K. Furukawa, Ed.), MIT Press, Cambridge, MA.

VAN EMDEN, M., AND KOWALSKI, R. A. (1976), The semantics of predicate logic as a programming language, *J. Assoc. Comput. Mach.* **23**, 733–742.

FALASCHI, M., GABBRIELLI, M., LEVI, G., AND MURAKAMI, M. (1990), Nested guarded Horn clauses: A language provided with a complete set of unfolding rules, *Intern. J. Found. Comput. Sci.* **1**(3), 249–263.

FALASCHI, M., AND LEVI, G. (1990), Finite failures and partial computations in concurrent logic languages, *Theoret. Comput. Sci.* **75**, 45–66.

FALASCHI, M., LEVI, G., MARTELLI, M., AND PALAMIDESSI, C. (1988), A new declarative semantics for logic languages, *in* "Proceedings, Fifth International Conference on Logic Programming" (K. Bowen and R. A. Kowalski, Eds.), pp. 993–1005, MIT Press.

FALASCHI, M., LEVI, G., MARTELLI, M., AND PALAMIDESSI, C. (1989), Declarative modeling of the operational behaviour of logic languages, *Theoret. Comput. Sci.* **69**(3), 289–318.

GABBRIELLI, M., AND LEVI, G. (1990), Unfolding and fixpoint semantics of concurrent constraint logic programs, *in* "Proceedings, Second International Conference on Algebraic and Logic Programming" (H. Kirchner and W. Wechler, Eds.), pp. 204–216, Lecture Notes in Computer Science, Vol. 463, Springer-Verlag, Berlin/New York.

GABBRIELLI, M., AND LEVI, G. (1991), Modeling answer constraints in constraint logic programs, *in* "Proceedings, Eight International Conference on Logic Programming" (K. Furukawa, Ed.), MIT Press, Cambridge, MA.

GAIFMAN, H., AND SHAPIRO, E. (1989a), Proof theory and semantics of logic programs, *in* "Proceedings, Fourth Annual Symposium on Logic In Computer Science," pp. 50–62, IEEE Comput. Soc. Press, New York.

GAIFMAN, H., AND SHAPIRO, E. (1989b), Fully abstract compositional semantics for logic programs, *in* "Proceedings, Sixteenth Annual ACM Symposium on Principles of Programming Languages," pp 134–142, ACM Press, New York.

GIACOBAZZI, R., AND RICCI, L. (1990), Pipeline optimizations in AND-parallelism by abstract interpretation, *in* "Proceedings, Seventh International Conference on Logic Programming" (D. H. D. Warren and P. Szeredi, Eds.), pp. 291–305, MIT Press, Cambridge, MA.

KEMP, R., AND RINGWOOD, G. (1990), An algebraic framework for the abstract interpretation of logic programs, *in* "Proceedings, North American Conference on Logic Programming 1990" (S. Debray and M. Hermenegildo, Eds.), pp. 506–520, MIT Press, Cambridge, MA.

LASSEZ, J. L., MAHER, M. J., AND MARRIOTT, K. (1987), Unification revised, *in* "Foundations of logic and functional programming" (M. Boscarol, L. Carlucci Aiello, and G. Levi, Eds.), pp. 67–113, Lecture Notes in Computer Science, Vol. 306, Springer-Verlag, Berlin/New York.

LEVI, G. (1988), Models, unfolding rules, and fixpoint semantics, *in* "Proceedings, Fifth International Conference on Logic Programming" (K. Bowen and R. A. Kowalski, Eds.), pp. 1649–1665, MIT Press, Cambridge, MA.

LEVI, G., AND MANCARELLA, P. (1988), "The Unfolding Semantics of Logic Programs," Technical Report TR-13-88, Dipartimento di Informatica, Università di Pisa.

LEVI, G., MARTELLI, M., AND PALAMIDESSI, C. (1990), Failure and success made symmetric, *in* "Proceedings, North American Conference on Logic Programming 1990" (S. Debray and M. Hermenegildo, Eds.), pp. 3–22, MIT Press, Cambridge, MA.

LEVI, G., AND PALAMIDESSI, C. (1987), An approach to the declarative semantics of synchronization of logic languages, *in* "Proceedings, Fourth International Conference on Logic Programming" (J.-L. Lassez, Ed.), pp. 877–893, MIT Press, Cambridge, MA.

LLOYD, J. W. (1987), "Foundations of Logic Programming," Springer-Verlag, Berlin/New York.

MAHER, M. J. (1987), Equivalences of logic programs, *in* "Foundations of Deductive Databases and Logic Programming" (J. Minker, Ed.), pp. 627–658, Morgan Kaufmann.

MURAKAMI, M. (1990), A declarative semantics of flat guarded horn clauses for programs with perpetual processes, *Theoret. Comput. Sci.* **75**, 67–83.

TURI, D. (1991), Extending S-models to logic programs with negation, *in* "Proceedings, Eight International Conference on Logic Programming" (K. Furukawa, Ed.), pp. 397–411, MIT Press, Cambridge, MA.