Logic Programming Semantics Made Easy

Els Laenens
Dirk Vermeir

Department of Computer Science
University of Antwerp (UIA)
Universiteitsplein 1, B2610 Wilrijk, Belgium

Carlo Zaniolo

MCC
3500 West Balcones Center Drive,
Austin, Texas 78759-6509, USA

ABSTRACT

We propose a new model-theoretic semantics for logic programs, called pure semantics, based on the notions of unfounded set and assumption set. The pure semantics emerges from the observation that major logic programming semantics have the following feature in common: given an 'intended model' M, the set of negative literals in M corresponds exactly with the greatest unfounded set w.r.t. the set of positive literals in M. In other words, a model contains redundant information as its negative part can be described in function of its positive part. Thus, the total models and the partial models of programs can now be characterized by a set of positive literals. Based on this idea, we develop the pure semantics for logic programs. The result is a remarkably simple semantics that unifies previous approaches and explains how partial model semantics follows from a weaker closed world assumption.

1. Introduction

This note presents a new model-theoretic semantics for logic programs, called pure semantics, based on the notions of unfounded set[Gel88a] and assumption set [Lae90a, Lae90b]. The logic programs under consideration are general logic programs[Gel89a] (or seminegative programs[Lae90c]), i.e. sets of rules of the form

$$L_1 , L_2 \dots , L_n \rightarrow A$$

where A is an atom and $L_1 , \dots , L_n$ are literals (atoms or negated atoms).

Given a rule r, $H(r)$ denotes the head of r and $B(r)$ denotes the set of all literals in the body of r. A term, atom, rule is *ground* if it is variable-free.

Given a logic program P, the Herbrand Universe of P (denoted H_P) is the set of all possible ground terms. The Herbrand Base of P (denoted $B[P]$) is the set of all possible ground atoms whose predicate symbols occur in P and whose arguments are elements of H_P. A *ground instance* of a rule r in P is a rule obtained from r by replacing every variable X in r by $\phi(X)$, where ϕ is a mapping from the set of all variables occurring in P to H_P. The set of all ground instances of all rules in P is denoted by $ground(P)$. Any subset of $B[P] \cup Not\ B[P]$ is called an *interpretation* of P if it is consistent, i.e. there are no two literals A and B in X such that $A = Not\ B$.

Notation. Let S be a set of ground literals. The set of atoms that occur positively, resp. negatively, in S is denoted $pos(S)$, resp. $neg(S)$. $Not\ S$ denotes the set $\{Not\ s \mid s \in S\}$. □

Definition [Gel88a]**.** Let P be a logic program and I an interpretation of P. $X \subseteq B[P]$ is an *unfounded set* of P w.r.t. I if for each $p \in X$ and for each $r \in ground(P)$ with $H(r)=p$ one of the following holds:

1. $B(r) \cap Not\ I \neq \emptyset$, i.e. r is *blocked* in I
2. $B(r) \cap X \neq \emptyset$.

The *greatest unfounded set* of P w.r.t. I, denoted $U(I)$, is the union of all sets that are unfounded sets of P w.r.t. I. $U(I)$ is easily seen to be an unfounded set. □

The pure semantics emerges from the observation that major logic programming semantics - such as the well-founded semantics [Gel88a], the stable partial model semantics [Sac90a], the three-valued stable semantics[Prz90a] - have the following feature in common: given an 'intended model' M, the set of negative literals in M corresponds exactly with the greatest unfounded set w.r.t. the set of positive literals in M, i.e. $neg(M)=U(pos(M))$. In other words, a model contains redundant information as its negative part can be described in function of its positive part. So we can follow the line of the stable model semantics[Gel88b] in the sense that a model is a set of atoms rather than a set of literals. Based on this idea, we next develop the pure semantics for logic programs. The result is a remarkably simple semantics that unifies previous approaches and explains how partial model semantics follows from a weaker closed world assumption.

The proofs of all results presented in the sequel can be found in [Lae91a].

2. The Pure Semantics

Definition. Let P be a logic program. A *positive interpretation* of P is any subset of $B[P]$. □

Intuitively, an unfounded set w.r.t. an interpretation I contains atoms that are known to be non-inferrable from I and from any extension of I. The next definition (which resembles the definition of unfounded sets) introduces assumption sets. Assumption sets differ from unfounded sets in that the atoms they contain are not inferrable from the interpretation at hand but might be inferrable from extensions of it. As an example, consider the logic program consisting of two rules $Not\ p \rightarrow q$ and $Not\ q \rightarrow p$; and let I be the empty set. The greatest unfounded set w.r.t. I is empty whereas $\{p,q\}$ is an assumption set as neither p nor q is inferrable from I. Formally, this amounts to the following definition.

Definition. Let P be a logic program and let I be a positive interpretation of P. $X \subseteq B[P]$ is an *assumption set* of P w.r.t. I if for each p in X and for each rule r in $ground(P)$ with $H(r)=p$, one of the following holds:

1. $B(r) \cap Not\ I \neq \varnothing$
2. $B(r) \cap X \neq \varnothing$
3. $B(r) \not\subseteq I \cup Not\ U(I)$, i.e. r is *non-applicable* in I

The *greatest assumption set* of P w.r.t. I, denoted $A(I)$, is the union of all sets that are assumption sets of P w.r.t. I. $A(I)$ is easily seen to be an assumption set.

I is said to be *assumption-free* iff $A(I) \cap I=\varnothing$. □

Since the notion of assumption set relaxes the notion of unfounded set by adding an alternative condition (3), we conclude:

Proposition 1. Given a logic program P and a positive interpretation I of P, every unfounded set of P w.r.t. I is an assumption set of P w.r.t. I. □

Let us take a closer look at the new condition in the above definition. Traditionally, a rule r is applicable in a (partial) interpretation I if $B(r) \subseteq I$ which means that the atoms occurring positively in r's body are *true* w.r.t. I, i.e. $pos(B(r)) \subseteq I$, and that the atoms occurring negatively in its body are *false* w.r.t. I, i.e. $neg(B(r)) \subseteq Not\ I$. It is our feeling that this (traditional) definition of false atoms (i.e. an atom p is false w.r.t. I iff $Not\ p \in I$) obscures what falsity is really about. Indeed, it is essential to remember that an atom is not false because its negation is in the interpretation, but that the negation of an atom is in the interpretation because the atom is found to be false. An excellent illustration of this is the fixpoint computation of the well-founded model in[Gel88a] where each iteration adds, besides the positive atoms as inferred using the immediate consequence transformation, also the negation of each atom in the greatest unfounded set which is the set of ground atoms that are already known to be false. This suggests that the greatest unfounded set w.r.t. an interpretation I is actually the set of all atoms that are false w.r.t. I. It follows that the classical definition of falsity only makes sense when $neg(I)=U(I)$. So condition 3 in the definition of assumption set uses a more exact definition of rule applicability, based on the essential meaning of falsity: an atom p is false w.r.t. I iff $p \in U(I)$. As we shall see later, this tantamounts to a more conservative application of the CWA.

Intuitively, the greatest assumption set w.r.t. a positive interpretation I is the set of all atoms that are non-inferrable from I. In view of this fact, whenever $I \cap A(I) \neq \varnothing$ we know that I contains 'assumptions', i.e. atoms for which no proper motivation of their presence in I exists. Accordingly, $\overline{A(I)}=B[P]-A(I)$ contains all atoms that are inferrable from I and consequently I is not deductively closed whenever $\overline{A(I)}-I \neq \varnothing$. It is evident that we want a model to be deductively closed and assumption-free at the same time. This leads us to the introduction of pure models.

Definition. Let P be a logic program and M a positive interpretation of P. M is a *pure (partial) model* of P iff its complement, $\overline{M}=B[P]-M$, is the greatest assumption set w.r.t. M, i.e. $\overline{M}=A(M)$. □

The following equivalence is useful in the proofs.

Proposition 2. Let P be a logic program and let M be a positive interpretation. M is a pure model iff it is assumption-free and its complement $\overline{M}$ is an assumption set w.r.t. M. □

Since pure models are assumption-free, they are - according to Proposition 1 - also free from 'unfounded literals'.

Proposition 3. Given a logic program P and a pure model M of P. Then $M \cap U(M)=\varnothing$. □

The pure semantics is universal in the sense that it captures the meaning of every logic program.

Theorem 1. Every logic program has at least one pure model.

Proof (sketch). $pos(W_P^{\infty}(\varnothing))$[Gel88a] is a pure model of P. □

3. Examples

Consider the logic program P_1.

$$\begin{array}{l} Not\ p \rightarrow q \\ Not\ q \rightarrow p \end{array}$$

Let $M_1=\varnothing$. Then $U(M_1)=\varnothing$. $\overline{M_1}=B[P_1]=\{p, q\}$ is the greatest assumption set w.r.t. M_1 as no rule for p or q is applicable in M_1. Hence, M_1 is a pure model of P_1.

Let $M_2=\{p\}$. Then $U(M_2)=\{q\}$. So $\overline{M_2}=\{q\}$ is the greatest unfounded set w.r.t. M_2, and since every unfounded set is an assumption set (Proposition 1), we find that $\overline{M_2}$ is an assumption set. Moreover, $\{p,q\}$ is not an assumption set as the rule $Not\ q \rightarrow p$ does not satisfy any of the conditions in the definition of assumption set: $Not\ q \rightarrow p$ is applicable in M_2, not blocked in M_2 and its body does not contain any 'assumptions'. So, $\overline{M_2}$ is the greatest assumption set w.r.t. M_2 which means that M_2 is a pure model of P_1. In much the same way, one can show that also $M_3=\{q\}$ is a pure model. And that's it; $M_4=\{p, q\}$ is not a pure model as the greatest assumption set w.r.t. M_4 is M_4 itself.

Consider P_2 consisting of only one rule.

$$Not\ p \rightarrow p$$

The empty set is the only pure model of P_2 since $B[P_2]=\{p\}$ is an assumption set w.r.t. $\varnothing$ (as $U(\varnothing)=\varnothing$) and also w.r.t. $\{p\}$.

The following logic program P_3 (two copies of P_1)

$$\begin{array}{l} Not\ p \rightarrow q \\ Not\ q \rightarrow p \\ Not\ a \rightarrow b \\ Not\ b \rightarrow a \end{array}$$

has nine pure models: $\varnothing$, $\{a\}$, $\{b\}$, $\{p\}$, $\{q\}$, $\{p,a\}$, $\{p, b\}$, $\{q,a\}$, and $\{q, b\}$.

Consider P_4.

$$\begin{array}{l} Not\ b \rightarrow a \\ Not\ a \rightarrow b \\ Not\ c \rightarrow a \\ Not\ a\ , Not\ b \rightarrow c \end{array}$$

This program has two pure models: $\{a\}$ and the empty set.

Consider P_5 (basically two copies of P_4).

$$\begin{array}{l} Not\ b \rightarrow a \\ Not\ a \rightarrow b \\ Not\ c \rightarrow a \\ Not\ a\ , Not\ b \rightarrow c \end{array}$$

$$\begin{array}{l} Not\ p \rightarrow q \\ Not\ q \rightarrow p \\ Not\ r \rightarrow q \\ Not\ q\ , Not\ p \rightarrow r \end{array}$$

$$a\ , q \rightarrow s$$

P_5 has four pure models: the empty set, $\{a\}$, $\{q\}$ and $\{a, q, s\}$.

4. Relation to Other Approaches

The stable model semantics[Gel88b] is not universal. Some logic programs such as P_2 do not have any stable model. The main difference between the stable and the pure semantics is that - although both are based on positive interpretations - they are actually two-valued and three-valued respectively. Indeed, a rule r is applicable in a stable model M if $pos(B(r)) \subseteq M$ and $neg(B(r)) \cap M=\varnothing$, i.e. $B(r) \subseteq M \cup Not\ \overline{M}$; whereas r is applicable in a pure model M if $B(r) \subseteq M \cup Not\ U(M)$. In other words, the set of atoms that are considered false w.r.t. M is $\overline{M}$ in the case of a stable model M and $U(M)$ in the case of a pure model M. So $B[P]-(M \cup U(M))=\overline{M}-U(M)$ is the set of atoms that are undefined w.r.t. a pure model M. It is then intuitively clear that the pure models that are two-valued or total (i.e. $M \cup U(M)=B[P]$ or $\overline{M}-U(M)=\varnothing$) are exactly the stable models.

Definition. Given a logic program P, a positive interpretation I of P is called *total* iff $I \cup U(I)=B[P]$. □

Theorem 2. The total pure models of a logic program P are identical to the stable models[Gel88b] of P.

Proof (sketch). This theorem follows immediately from Theorem 4 (b) (which is proved independently). □

Further relationships between the pure model semantics and other model-theoretic approaches will be described in terms of 'equivalences' between interpretations and positive interpretations.

Definition. Let P be a logic program. We say that a positive interpretation A of P is *equivalent* with an interpretation I of P iff $I = A \cup Not\ U(A)$. □

Theorem 3. Let P be a logic program. The pure models of P are equivalent with the three-valued stable models[Prz90a] of P, i.e.

(a) If M is a three-valued stable model of P, then $pos(M)$ is a pure model of P; and

(b) If M is a pure model of P, then $M \cup Not\ U(M)$ is a three-valued stable model of P.

Proof (sketch). (a) One can show[Lae91a] that three-valued stable models of P are fixpoints of W_P and that assumption-free fixpoints of W_P are equivalent with pure models of P. So, if $pos(M)$ is assumption-free, then we can infer that it is a pure model of P. It is easy to see that $M-A(pos(M))$ is a model of P/M. But, by definition, M is the least model of P/M so $M \cap A(pos(M)) = \emptyset$. Hence, $pos(M)$ is indeed assumption-free and part (a) of the theorem follows.

(b) It is straightforward to show that $M \cup Not\ U(M)$ is a three-valued model[Prz89a] of $P/(M \cup Not\ U(M))$.

Suppose that M' is the least three-valued model of $P/(M \cup Not\ U(M)$. Then $M' \subseteq M \cup Not\ U(M)$. One can show that for each $p \in M-M'$ and for each rule r in $ground(P)$ with $H(r)=p$ either $B(r) \not\subseteq M \cup Not\ U(M)$, or $B(r) \cap Not\ M \neq \emptyset$, or $B(r) \cap (M-M') \neq \emptyset$. In other words, $M-M'$ is an assumption set w.r.t. M. So $M-M'$ must be empty as M is a pure model and thus assumption-free. Therefore $M=pos(M')$. Moreover, as M' is a three-valued stable model of P, it is a fixpoint of W_P and thus $M'=pos(M') \cup Not\ U(pos(M'))$ [Lae91a]. So, we find $M'=M \cup Not\ U(M)$ which proves part (b) of the theorem. □

This result is significant, since it clarifies the significance of the three-valued model proposed in[Prz90a] as follows:

- complications, such as three-valued logic and negative literals in the model, are unnecessary, and, in fact obscure, the fundamental conceptual shift described next.
- the transition from total stable models to partial stable models tantamounts to the relaxation of the CWA, whereby only atoms that belong to the greatest unfounded set are now assumed to be false.

We now consider for a given logic program P the partially-ordered set of pure models of P where " $\subseteq$ " (set inclusion) is the partial order, and turn our attention to its maximal for P. Indeed, the notions justifiable for regular models and founded for stable models are equivalent and clearly the principle of minimal undefinedness for regular models corresponds to the maximality of stable partial models.

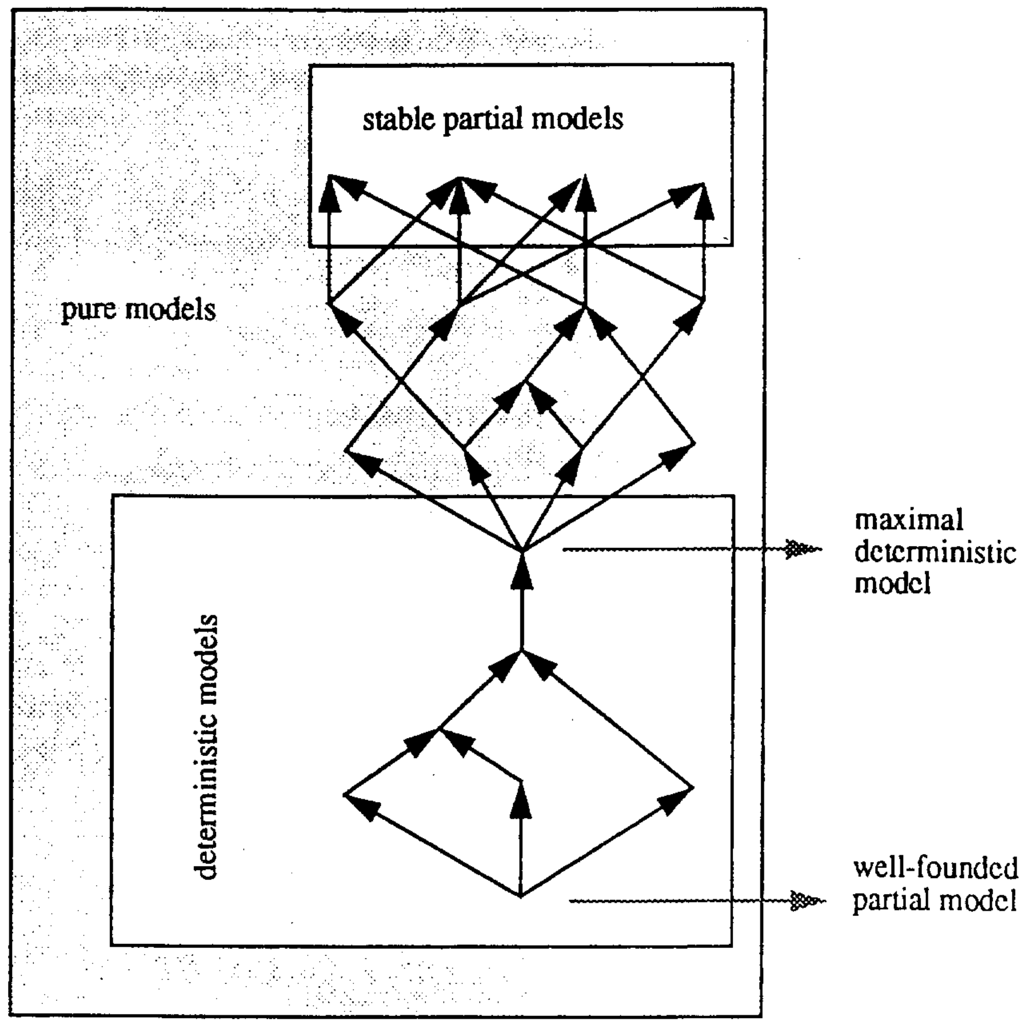

Fig. 3.

(c) In[Lae91a] we show that the greatest lower bound in $< \mu , \subseteq >$ of all maximal pure models of P exists, by proving that for every two pure models, M_1 and M_2, that are contained in the deterministic set[Sac90a], (i.e. the intersection of all maximal pure models), $pos(W_P^{\infty}(M_1 \cup M_2))$ is again a pure model that contains $M_1 \cup M_2$ and which is a subset of the deterministic set. So, the glb G of all maximal pure models is the maximal pure model that is contained in the deterministic set. Since the pure models of P are equivalent with the strongly-founded models of P, G is equivalent with the maximal strongly-founded model that is contained in the deterministic set, which is exactly the definition of the maximal deterministic model. □

So, $< \mu , \subseteq >$ provides us with a dag structure (directed acyclic graph) of which

- the root, i.e. unique node without incoming arcs, represents the well-founded model,
- the leafs, i.e. nodes without outgoing arcs, represent the stable partial models,
- the glb of the leafs represents the maximal deterministic model, and
- the 'ancestors' of the glb of the leafs represent the deterministic models.

This is depicted in Figure 3.

From Figures 1 and 2 we learn for instance that:

- P_3 has four stable partial models; its unique deterministic model is the well-founded model;
- P_5 has a single stable partial model which therefore coincides with its maximal deterministic model; its well-founded model is a different deterministic model;
- etc.

5. Conclusion

Much of the current research aims to provide semantics which is universal and deals effectively with incompleteness. The contribution of this paper is to simplify and unify previous approaches by eliminating the need for three-valued logic and negated literals in defining partial models. Moreover, it shows that the conceptual transition from total model semantics to partial model semantics boils down to a simple relaxation of the CWA.

The 'purification' of traditional logic programming semantics as presented in this paper also clarifies the meaning of negation as failure. In[Lae92a], the pure semantics for extended logic programs is shown to be a natural extension of the pure semantics for general logic programs.

6. Acknowledgment

The first author would like to thank Natraj Arni for interesting discussions.

References

Gel88a.
A. Van Gelder, K. Ross, and J. S. Schlipf, *The Well-Founded Semantics for General Logic Programs*, 1988. UCSC-CRL-88-16

Gel88b.
M. Gelfond and V. Lifschitz, "The Stable Model Semantics for Logic Programming," in *Proc. of the Intl. Conf. on Logic Programming*, pp. 1071-1079, 1988.

Gel89a.
M. Gelfond and V. Lifschitz, *Logic Programs with Classical Negation*, 1989. Manuscript

Lae90a.
E. Laenens and D. Vermeir, "A Fixpoint Semantics of Ordered Logic," *Journal of Logic and Computation*, vol. 1, no. 2, pp. 159-185, 1990.

Lae90b.
E. Laenens, ''Foundations of Ordered Logic,'' *PhD Thesis, University of Antwerp (UIA)*, 1990.

Lae90c.
E. Laenens, D. Vermeir, and D. Sacca, ''Extending logic programming,'' in *Proceedings of the SIGMOD conference*, pp. 184-193, 1990.

Lae91a.
E. Laenens, D. Vermeir, and C. Zaniolo, ''Pure models for logic programs: a simplification and unification of logic programming semantics,'' *Technical report, 91-31*, 1991.

Lae92a.
E. Laenens, *A universally consistent semantics for extended logic programs*, Submitted for puclication, 1992.

Prz89a.
T. Przymusinski, ''Three-Valued Formalizations of Non-Monotonic Reasoning and Logic Programming,'' *Proc. 1st Int. Conference on Principles of Knowledge Representation and Reasoning*, pp. 341-349, 1989.

Prz90a.
T. Przymusinski, ''Well-founded semantics coincides with three-valued stable semantics,'' *Fundamenta Informaticae*, vol. 13, pp. 445-463, IOS Press, 1990.

Sac90a.
D. Sacca and C. Zaniolo, ''Stable models and Non-determinism for logic programs with negation,'' *Proc. ACM Symp. on Principles of Database Systems*, 1990.

You90a.
Jia-Huai You and Li Yan Yuan, ''Three-Valued Formalization of Logic Programming: Is It Needed?,'' in *Proc. of the PODS'90 conference*, pp. 172-182, 1990.