URL: http://www.elsevier.nl/locate/entcs/volume40.html 17 pages

On the Coincidence of Semantics for Uniquely Determined Programs

Pascal Hitzler [1,2]

Artificial Intelligence Institute
Department of Computer Science
Dresden University of Technology
Dresden, Germany

Anthony Karel Seda [3]

Department of Mathematics
National University of Ireland, University College Cork
Cork, Ireland

Abstract

We study classes of logic programs, called here *unique supported model classes*, or simply *usm-classes*, with the property that each member in the class is uniquely determined, that is, possesses a unique supported model. Known classes of uniquely determined programs include the acyclic and the acceptable programs, which have been much studied in the context of termination, and the authors gave a unifying treatment of these and other unique supported model classes in an earlier paper. In this paper, we complement these earlier results by considering how various standard semantics relate to each other within certain unique supported model classes. In particular, we introduce the natural usm-class of all Φ-accessible programs, which contains the aforementioned classes, and has the property that, for each member of it, the stable, well-founded and weakly perfect-a models all coincide.

1 Introduction

A logic program P is a (not necessarily finite) set of (universally quantified) clauses, or axioms, $A \leftarrow L_1, \ldots, L_n$, sometimes abbreviated to $A \leftarrow \texttt{body}$,

[1] The first named author acknowledges financial support under grant SC/98/621 from Enterprise Ireland.

[2] Email: phitzler@inf.tu-dresden.de,
Homepage: http://www.wv.inf.tu-dresden.de/~pascal/

[3] Email: a.seda@ucc.ie, Homepage: http://maths.ucc.ie/staff/seda/

where the atom A and the literals L_i are drawn from some given first order language $\mathcal{L}$ (often referred to as the underlying language of P), and $L_1, \ldots, L_n$ denotes the conjunction $L_1 \wedge \ldots \wedge L_n$. A logic programming system can be thought of as a simple model of reasoning, since computation by a program P within it is the derivation of logical consequences from the program P. While current implementations of such systems, including the Prolog family and answer set programming systems [17], suffer from drawbacks which restrict the syntax of allowable programs, it is nevertheless worthwhile studying logic programs quite generally in order to better understand reasoning *per se* within this computational paradigm.

Thinking of a logic program as a set of axioms in first order logic immediately gives rise to a declarative reading in the form of its models in the usual sense of mathematical logic. In the theory of logic programming, however, not all of these models are considered satisfactory. Rather, one seeks to distinguish models with certain properties with a view to capturing the *intended meaning* of the programmer or to mirror the abilities of the logic programming system on which the program is run.

A very natural way of assigning meaning to logic programs is via the supported model semantics or Clark completion semantics of [7] which has good properties relative to *negation as failure* (familiar when implementing Prolog systems by means of resolution). However, in contrast with semantical approaches such as the well-founded [11] or the weakly perfect model semantics [20], a program can have several different meanings under the supported model semantics — a property which it shares with its refinement, the stable model semantics [12]. Whether this is a weakness or a feature is in fact a matter of point of view: answer set programming systems make explicit use of the ambiguity of the stable model semantics, and this is discussed in [17].

Nevertheless, some classes of programs which have been discussed in the literature have the property that they are *uniquely determined* in that each program in the class has a *unique* supported model. In this paper, such classes will be called *unique supported model classes*. Since other two-valued semantics can be understood as refinements of the supported model semantics, it may be expected that the supported model semantics coincides with these other declarative readings in the case of uniquely determined programs. Indeed, we will see in the sequel that this is true for the particular unique supported model class of Φ-*accessible* programs, due to the authors, but does not carry over in full generality to all uniquely determined programs.

Unique supported model classes include some programs which have been studied in the context of termination, such as the acyclic or the acceptable programs [2], and straightforward generalizations of these, for example locally hierarchical programs [6]. In [14], a unifying approach to some unique supported model classes using operators in three-valued logics was introduced which, amongst other things, led to the definition of the unique supported model class of all Φ^*-accessible programs, a subclass of the Φ-accessible programs. This class is remarkable since it is computationally adequate in the sense that every partial recursive function can be implemented, under Prolog, by such a program. This does not hold for the subclass of acceptable programs which always terminate under the Prolog selection rule. By studying unique supported model classes one is therefore working in a setting which is semantically relatively unambiguous, and one may still study classes of programs which range from those having strong termination properties to those having full computational adequacy. The approach in [14] mentioned earlier led to a comparison of several unique supported model classes and the determination of some of their distinguishing properties. In this paper, we intend to complement these earlier results and discuss how unique supported model classes fit into the framework given by other classes of programs and other standard approaches to semantics.

The plan of the paper is as follows. After some preliminaries in Section 2, where we will recall the definition of the Φ^*-accessible programs and define the larger class of Φ-accessible programs, we will give alternative characterizations of these in Section 3. The new characterizations which are obtained are in the spirit of the definition of the acceptable programs due to Apt and Pedreschi [2]. In Section 4, we will see that the natural semantics of these programs can be obtained as the limit, in the atomic topology, of iterates of the single-step operator without having to resort to operators in three-valued logic. A generalized metric approach to the same problem will also be treated briefly. In Section 5, we study Φ-accessible programs in the context of weak stratification. We will see that for Φ-accessible programs, the supported, minimal, well-founded, weakly perfect, and stable model semantics all coincide. Section 6 then focusses on the relationship between unique supported model classes and the stable model semantics. While existence and uniqueness of a supported model does not guarantee the existence of a stable model, we will be able to describe a class of programs, using our earlier results, for which the supported models are exactly the stable models. This result again uses the notion of Φ^*-accessible program. We will finally, in Section 7, discuss some conclusions which may be drawn from the results presented here.

2 Preliminaries

We study normal logic programs and work over Herbrand preinterpretations only. Many of our results carry over easily to more general settings, see [13]. Thus, B_P denotes the Herbrand base for a given program P, and I_P denotes the power set of B_P which can be identified, as usual, with the set of all (Herbrand-) interpretations for P. Our reference to standard concepts and notation in logic programming is [16]. By ground(P) we will denote the program consisting of all ground instances of clauses in P. A level mapping for a program P is a mapping $l : B_P \to \alpha$, where α is a countable ordinal. We always automatically extend l to ground literals by defining $l(\neg A) = l(A)$ for

Table 1
Truth table for Kleene's strong three-valued logic.

p	q	$p \wedge q$	$\neg p$	p	q	$p \wedge q$	$\neg p$	p	q	$p \wedge q$	$\neg p$
t	t	t	f	u	t	u	u	f	t	f	t
t	u	u	f	u	u	u	u	f	u	f	t
t	f	f	f	u	f	f	u	f	f	f	t

all ground atoms A.

We assume that the reader is familiar with the perfect model semantics [21], the stable model semantics [12] and the well-founded model semantics [11] for normal logic programs. Given a program P, the *supported models* for P are the fixed points of the single-step operator $T_P : I_P \to I_P$. As usual, $T_P(I)$ is defined to be the set of all $A \in B_P$ such that there exists a clause $A \leftarrow$ `body` in ground(P) with the property that `body` is true in I. Recall that supported models can be identified with the models of the Clark-completion of P, see [7]. We also assume that the reader is familiar with the notion of weakly stratified program, see [5,20].

In Section 4, we employ the notion of convergence in the atomic topology in order to obtain models for some unique supported model classes. The atomic topology was introduced in [23]. In the Herbrand case, it coincides with the query topology of [4].

Definition 2.1 The *atomic topology* Q on the space I_P of all interpretations is characterized via convergence as follows: a net[4] (I_λ) of interpretations converges in Q if and only if, for every $A \in B_P$, there exists an index α such that either $I_\beta \models A$ for all $\beta > \alpha$ or for all $\beta > \alpha$ we have $I_\beta \not\models A$, that is, iff every $A \in B_P$ is either eventually true in I_λ or is eventually false in I_λ. The unique limit of a converging net is the interpretation which assigns the truth value *true* to all ground atoms which are eventually true in I_λ.

Following [10], a *partial* (or *three-valued*) *interpretation* I is a pair (I^+, I^-) of subsets of B_P such that I^+ and I^- are disjoint. Partial interpretations are interpreted in Kleene's strong three-valued logic: given a partial interpretation $I = (I^+, I^-)$, atoms in I^+ carry the truth value *true* (t) in I and atoms in I^- the value *false* (f) in I. Atoms which are neither in I^+ nor in I^- carry the truth value *undefined* (u). For our purposes, we only need to know how conjunction and negation operate in this three-valued logic, and this is given by the truth table in Table 1.

A partial interpretation (I^+, I^-) is called *total* if $I^+ \cup I^- = B_P$, and such

[4] The topological notion of a net (see [24] or any book on general topology) is necessary in order to characterize Q if the domain of the preinterpretation is uncountable. When dealing with Herbrand preinterpretations, sequences suffice. However, we will need to work with transfinite sequences later on, which are a special case of nets more general than sequences.

interpretations can be naturally identified with elements of I_P. The set $I_{P,3}$ of all partial interpretations is a complete partial order, indeed complete semi-lattice, under the ordering $(I_1^+, I_1^-) \leq (I_2^+, I_2^-)$ iff $I_1^+ \subseteq I_2^+$ and $I_1^- \subseteq I_2^-$, with bottom element $\bot = (\emptyset, \emptyset)$. Total interpretations are in fact maximal elements in the given ordering.

The three-valued operator Φ_P (cf. [10]) is defined as a mapping on partial interpretations K as follows. We set $\Phi_P(K) = (I^+, I^-)$, where I^+ is the set of all $A \in B_P$ with the property that there exists a clause $A \leftarrow \texttt{body}$ in ground(P) such that $\texttt{body}$ is true in K, and I^- is the set of all $A \in B_P$ such that for all clauses $A \leftarrow \texttt{body}$ in ground(P) we have that $\texttt{body}$ is false in K; truth and falsehood being taken here relative to Kleene's strong three-valued logic. We note that Φ_P is monotonic, and hence we define

$$
\begin{aligned}
\Phi_P\!\uparrow\!0 &= \bot, \\
\Phi_P\!\uparrow\!(k+1) &= \Phi_P(\Phi_P\!\uparrow\!k) \text{ for any ordinal } k, \text{ and} \\
\Phi_P\!\uparrow\!\alpha &= \sup\{\Phi_P\!\uparrow\!\beta \mid \beta < \alpha\} \text{ for any limit ordinal } \alpha.
\end{aligned}
$$

The three-valued operator Φ_P^* is defined as a mapping on partial interpretations K as follows. We set $\Phi_P^*(K) = (I^+, I^-)$, where I^+ is the set of all $A \in B_P$ with the property that for all clauses $A \leftarrow \texttt{body}$ in ground(P) we have that $\texttt{body}$ is not undefined in K and at least one of these bodies is true in K, and I^- is the set of all $A \in B_P$ such that for all clauses $A \leftarrow \texttt{body}$ in ground(P) we have that $\texttt{body}$ is false in K. This operator is also monotonic and its ordinal powers are defined as for Φ_P above; it was studied in more detail in [14] where Φ_P and Φ_P^* were denoted by $\Phi_{P,1}$ and $\Phi_{P^*,1}$, respectively.

Definition 2.2 A normal logic program P is called Φ-*accessible*, respectively Φ^*-*accessible*, if there exists an ordinal α such that $\Phi_P\!\uparrow\!\alpha$, respectively $\Phi_P^*\!\uparrow\!\alpha$, is a total interpretation. The smallest ordinal with the first property is called the *closure ordinal* of P with respect to Φ, and the smallest ordinal with the second property is called the *closure ordinal* of P with respect to Φ^*.

As observed in [14], each acceptable [2], acyclic [6], or locally hierarchical [6] program is Φ^*-accessible, each Φ^*-accessible program is Φ-accessible, and each Φ-accessible program has a unique supported model. Indeed, in [14], we described more variations of the operator Φ, each of which gives rise to a unique supported model class analogous to the Φ-accessible programs. Of all the classes described in [14], the Φ-accessible programs form the largest.

3 Φ-accessible and Φ^*-accessible programs

Following the definition of acceptable programs [2], we define next two super-classes of these denoted by $[\Phi]$, respectively $[\Phi^*]$. As it turns out, these classes contain exactly the Φ-accessible, respectively Φ^*-accessible, programs.

Definition 3.1 Let P be a normal logic program. Then P is contained in $[\Phi]$ if and only if there exists a level mapping l for P and a (two-valued) model

I for P such that the following condition holds. Each $A \in B_P$ satisfies either (i) or (ii):

(i) There exists a clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) with head A such that $I \models L_1 \wedge \ldots \wedge L_n$ and $l(A) > l(L_i)$ for all $i = 1, \ldots, n$.

(ii) For each clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) with head A there exists $i \in \{1, \ldots, n\}$ such that $I \not\models L_i$, $I \not\models A$ and $l(A) > l(L_i)$.

To define $[\Phi^*]$: a program P is contained in $[\Phi^*]$ if and only if there exists a level mapping l for P and a (two-valued) model I for P such that the following condition holds. For each clause $A \leftarrow L_1, \ldots, L_n$ in ground(P), we either have $I \models L_1 \wedge \ldots \wedge L_n$ and $l(A) > l(L_i)$ for all $i = 1, \ldots n$ or there exists $i \in \{1, \ldots, n\}$ such that $I \not\models L_i$, $I \not\models A$ and $l(A) > l(L_i)$.

For every Φ-accessible program, we define a canonical level mapping as follows.

Definition 3.2 Let P be Φ-accessible. For each $A \in B_P$, let $l_P(A)$ denote the least ordinal α such that A is not undefined in $\Phi_P\uparrow(\alpha+1)$. We call the resulting mapping l_P the *canonical level mapping for P with respect to Φ.*

Theorem 3.3 *The class $[\Phi]$ contains exactly the Φ-accessible programs.*

Proof. Let P be Φ-accessible, let l_P be its canonical level mapping with respect to Φ, let α be its closure ordinal with respect to Φ and let $M_P = \Phi\uparrow\alpha^+$ be its unique supported (two-valued) model.
(a) Let $A \in M_P$ and let $l_P(A) = \beta$. By definition of l_P and Φ_P, there exists a clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) such that the $L_1, \ldots, L_n$ are true in $\Phi\uparrow\beta$ and, hence, are also true in M_P. Again by definition of l_P, we obtain $l_P(A) > l_P(L_i)$ for all i.
(b) Let $A \notin M_P$ and let $l_P(A) = \beta$. By definition of l_P and Φ_P, we obtain that for any clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) we must have that $L_1 \wedge \ldots \wedge L_n$ is false in $\Phi_P\uparrow\beta$. So, there must be some i such that L_i is false in $\Phi_P\uparrow\beta$ and $l_P(L_i) < \beta$ by definition of l_P, and hence $l_P(A) > l_P(L_i)$. Thus, $P \in [\Phi]$.

Conversely, let $P \in [\Phi]$ so that P satisfies conditions (i) and (ii) of Definition 3.1 with respect to a model I and a level mapping l. We show by induction on β that any $A \in B_P$ with $l(A) = \beta$ is not undefined in $\Phi_P\uparrow(\beta+1)$ and, furthermore, that I and $\Phi_P\uparrow(\beta+1)$ agree on A.
If $l(A) = 0$, then A must be the head of a unit clause or does not appear in any head. In the first case, A is true in $\Phi_P\uparrow 1$, and in the second case, A is false in $\Phi_P\uparrow 1$. Note that in the first case A is also true in I since condition (i) of Definition 3.1 applies and I is a model of P. Also, in the second case, A is also false in I since condition (ii) of Definition 3.1 applies.
Now let $l(A) = \beta$. If there is no clause in ground(P) with head A, then A is false in $\Phi_P\uparrow 1 \leq \Phi_P\uparrow(\beta+1)$ and also false in I since condition (ii) of Definition 3.1 applies. So assume there is a clause in ground(P) with head A. By definition of $[\Phi]$, either condition (i) or condition (ii) of Definition 3.1

applies.

If condition (i) applies, then there is a clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) such that $l(L_1), \ldots, l(L_n) < l(A)$ and therefore, by the induction hypothesis, the $L_1, \ldots, L_n$ are not undefined in $\Phi_P \uparrow \beta$ and I agrees with $\Phi_P \uparrow \beta$ on them. Now, since I is a model of P and $I \models L_1, \ldots, L_n$, we obtain that A is true in I and by definition of Φ_P also in $\Phi_P \uparrow \beta$.

If condition (ii) applies, then for each clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) there is some i such that $l(A) > l(L_i)$ and L_i is false in I. Hence we obtain that L_i is false in $\Phi_P \uparrow \beta$ by the induction hypothesis and it follows that A is false in both I and $\Phi_P \uparrow (\beta + 1)$. □

For Φ^*-accessible programs, we can derive a characterization along the very same lines as for Φ-accessible programs.

Definition 3.4 Let P be Φ^*-accessible. For each $A \in B_P$, let $l_P(A)$ be the least ordinal α such that A is not undefined in $\Phi^*_P \uparrow (\alpha + 1)$. We call l_P the *canonical level mapping for P with respect to Φ^*.*

Theorem 3.5 *The class $[\Phi^*]$ contains exactly the Φ^*-accessible programs.*

Proof. The proof is analogous to the proof of Theorem 3.3 and is omitted. □

Theorem 3.6 *Let P be Φ-accessible with unique supported model M. Then M is minimal as a two-valued model.*

Proof. Let $K \subseteq M$ be a model of P, and let l be the canonical level mapping of P with respect to Φ. Assume that there exists some $A \in M \setminus K$. Without loss of generality, we can assume that A is chosen such that $l(A)$ is minimal. By Theorem 3.3 and Definition 3.1, we obtain that there is a clause $A \leftarrow B_1, \ldots, B_k, \neg B_{k+1}, \ldots, \neg B_m$ in ground(P) with head A and $l(B_i) < l(A)$ for all atoms B_i in the body. Since $B_{k+1}, \ldots, B_m \notin M$, we obtain $B_{k+1}, \ldots, B_m \notin K$. By minimality of $l(A)$, we also obtain $B_1, \ldots, B_k \in K$. Now, since K is a model of P, we must have $A \in K$, which is a contradiction to our assumption. □

Let USM be the collection of all programs which have a unique supported model. In particular, $[\Phi] \subseteq$ USM.

Theorem 3.6 cannot be generalized to all programs in USM: the program

$$q \leftarrow p$$
$$p \leftarrow p, q$$
$$p \leftarrow \neg p, \neg q$$

has a unique supported model $\{p, q\}$, but $\{q\}$ is also a model (though not supported), and so $\{p, q\}$ is not minimal as a two-valued model.

Note also that for Φ^*-accessible programs the unique supported model is in general not least as a two-valued model as can be seen from the program consisting of the single clause $p \leftarrow \neg q$.

Theorem 3.7 *The definite programs in $[\Phi]$ are exactly the definite programs in USM.*

Proof. This follows immediately from [10, Proposition 7.3]: for a definite program P with least fixed point (I^+, I^-) of Φ_P, both I^+ and $B_P \setminus I^-$ are fixed points of the single-step operator T_P, and in fact I^+ is the least and $B_P \setminus I^-$ is the greatest supported model of P. Since P has only one supported model we obtain $I^+ = B_P \setminus I^-$ and therefore $P \in [\Phi]$. □

4 Topological Considerations

In this section, we will show that the unique supported model of a Φ-accessible program P can be obtained by applying the simpler single-step operator T_P. We recall the following result from [13, Corollary 4.2].

Proposition 4.1 *Let $I_n = T_P^n(\emptyset)$ and let $K_n = \Phi_P\!\uparrow n(\bot)$. Then, for all $n \in N$, we obtain $K_n^+ \subseteq I_n \subseteq {}^cK_n^-$, where ${}^cK_n^-$ denotes the complement $B_P \setminus K_n^-$ of K_n^-.*

We also recall from [2] (or [13] for arbitrary preinterpretations) that a total three-valued model (I^+, I^-) is a fixed point of Φ_P if and only if I^+ is a supported model.

Let us now define ordinal powers of the single-step operator T_P. Given a program P and an ordinal α, define $T_P\!\uparrow 0 = \emptyset$,

$$T_P\!\uparrow\alpha = \{A \in B_P \mid A \text{ is eventually contained in } (T_P\!\uparrow\beta(\emptyset))_{\beta<\alpha}\}$$

if α is a limit ordinal, and

$$T_P\!\uparrow\alpha = T_P(T_P\!\uparrow(\alpha - 1))$$

if α is a successor ordinal.

A similar construction was given in [3, Section 5.7], but the process given there closed off at the first infinite ordinal ω whereas this one need not do that. We note that this definition collapses to the usual one in the case of definite programs, see [16], but applies generally.

Theorem 4.2 *Let P be Φ-accessible and let $M = \Phi_P \uparrow \alpha$ be the least fixed point of Φ_P. Then, for all $\beta < \alpha$, we have $\Phi_P \uparrow \beta^+ \subseteq T_P \uparrow \beta \subseteq {}^c\Phi_P \uparrow \beta^-$. Furthermore, $(T_P\!\uparrow\beta)_{\beta<\alpha}$ converges in Q to $M^+ = T_P\!\uparrow\alpha$ which is the unique supported model of P.*

Proof. The first statement follows by an easy transfinite induction argument using Proposition 4.1. Since $\Phi_P\!\uparrow\alpha$ is total, the sequence $({}^c\Phi_P\!\uparrow\beta^- \setminus \Phi_P\!\uparrow\beta^+)_\beta$ converges in Q to $\emptyset$. With this observation, convergence of $(T_P\!\uparrow\beta)_{\beta<\alpha}$ follows immediately from the first statement. □

The result of Theorem 4.2 cannot be generalized to the larger class USM. Consider the following program P.

$p \leftarrow \neg q$
$q \leftarrow \neg p$
$p \leftarrow \neg p$

Then $\{p\}$ is the unique supported model of P. However, $T_P(\emptyset) = \{p, q\}$ and $T_P(\{p, q\}) = \emptyset$, so iterates of T_P do not converge in Q.

We can also derive the following very general theorem which generalizes [3, Lemma 38].

Theorem 4.3 *Let P be a normal logic program, let α be a limit ordinal, and assume that $(T_P\uparrow\beta)_{\beta<\alpha}$ converges in Q. Then $T_P\uparrow\alpha$ is a model of P.*

Proof. Suppose that $(T_P \uparrow \beta)_{\beta<\alpha}$ converges in Q to some M. We have to show that $T_P(M) \subseteq M$. Let $A \in T_P(M)$. Then there is a ground clause with head A and body B such that B is true in M. By convergence in Q, there exists an ordinal γ such that each of the literals in B is true with respect to $T_P\uparrow\delta$ for all $\delta \geq \gamma$. By definition of T_P and the existence of the above clause, we must therefore have $A \in T_P\uparrow\delta$ for all $\delta \geq \gamma + 1$, and therefore $A \in M$. □

The following corollary follows immediately.

Corollary 4.4 *The class of programs for which a model can be obtained as the limit in Q of transfinite iterates of the single-step operator contains the class of Φ-accessible programs.*

It is interesting to note that we can also cast the space I_P of all interpretations into a generalized ultrametric space, which allows the application of a fixed-point theorem due to Priess-Crampe and Ribenboim (see [15,18]) and we close this section by briefly considering this next.

Let P be Φ-accessible with unique supported model I and level mapping $l : I_P \to \gamma$. Let $\Gamma = \{2^{-\alpha} \mid \alpha \leq \gamma\}$, ordered by $2^{-\alpha} < 2^{-\beta}$ iff $\beta < \alpha$ and denote $2^{-\gamma}$ by 0. Thus, Γ is essentially $\gamma + 1$ endowed with the reverse order, but for historical reasons we prefer to work with the set Γ, see [15]. Define a function $d : I_P \times I_P \to \Gamma$ by setting $d(J, J) = 0$, and letting $d(J, K)$ equal $2^{-\alpha}$ provided that J and K differ on some ground atom of level α but agree on all ground atoms of lower level. Finally, define the function $\varrho : I_P \times I_P \to \Gamma$ by

$$\varrho(J, K) = \begin{cases} \max\{d(J, I), d(K, I)\} & \text{if } J \neq K \\ 0 & \text{if J=K.} \end{cases}$$

Proposition 4.5 *The space (I_P, ϱ) is a generalized ultrametric space. In other words, it satisfies the following conditions for all $I_1, I_2, I_3 \in I_P$.*

(Ui) $\varrho(I_1, I_2) = 0$ *implies* $I_1 = I_2$.

(Uii) $\varrho(I_1, I_1) = 0$.

(Uiii) $\varrho(I_1, I_2) = \varrho(I_2, I_1)$.

(Uiv) If $\varrho(I_1, I_2) \leq 2^{-\alpha}$ *and* $\varrho(I_2, I_3) \leq 2^{-\alpha}$, *then* $\varrho(I_1, I_3) \leq 2^{-\alpha}$.

Furthermore, (I_P, ϱ) is spherically complete, that is, the intersection of each chain[5] *of balls is non-empty. In this context, balls are sets of the form*

[5] By a chain of balls we mean a chain with respect to set-inclusion.

$B_\alpha(J) = \{K \in I_P \mid \varrho(J,K) \le 2^{-\alpha}\}$, *and* J *is called a* centre *of the ball (noting that any point of a ball is its centre) and* $2^{-\alpha}$ *its* radius.

Proof. (Ui) $\varrho(J,K) = 0$ obviously implies $J = K$. Also, (Uii), (Uiii) and (Uiv) are immediate.

For spherical completeness, let $\mathcal{B}$ be a chain of balls in I_P. If $\mathcal{B}$ contains a ball consisting of a single point, J, say, then it is clear that J belongs to every ball in the chain, and we are finished. So suppose every ball B in $\mathcal{B}$ contains at least two points; then $B = B_\alpha(M_B) = \{K \in I_P \mid \varrho(M_B,K) \le 2^{-\alpha}\}$ for some choice of centre M_B and α. Since $\varrho(M_B,K) \le 2^{-\alpha}$ implies $d(M_B,I) \le 2^{-\alpha}$ and $d(I,I) = 0$, it is now immediate from the definition of ϱ that $I \in B$. Thus, the intersection of $\mathcal{B}$ contains I and is, hence, non-empty, as required.□

Proposition 4.6 *Let* P *be* Φ*-accessible. Then* T_P *is strictly contracting with respect to* ϱ *in that it satisfies the condition* $\varrho(T_P(J), T_P(K)) < \varrho(J,K)$ *for all* $J,K \in I_P$.

Proof. Let $J,K \in I_P$ and assume that $\varrho(J,K) = 2^{-\alpha}$. Then J,K,I agree on all ground atoms of level less than α. We show that $T_P(J)$ and I agree on all ground atoms of level less than or equal to α. A similar argument shows that $T_P(K)$ and I agree on all ground atoms of level less than or equal to α, and this suffices.

Let $A \in T_P(J)$ with $l(A) \le \alpha$. Then there must be a clause $A \leftarrow L_1,\dots,L_n$ in ground(P) such that $J \models L_1 \wedge \dots \wedge L_n$. Since I and J agree on all ground atoms of level less than α, condition (ii) of Definition 3.1 cannot hold, because if $I \not\models L_i$ with $l(A) > l(L_i)$, then $J \not\models L_i$ and consequently $J \not\models L_1 \wedge \dots \wedge L_n$, a contradiction. Therefore, condition (i) of Definition 3.1 holds and so $A \in T_P(I) = I$. Hence, $A \in I$.

Conversely, suppose that $A \in I$. Since $I = T_P(I)$, there must be a clause $A \leftarrow L_1,\dots,L_n$ in ground(P) such that $I \models L_1 \wedge \dots \wedge L_n$. Thus, condition (i) of Definition 3.1 must hold, and so we can assume that $A \leftarrow L_1,\dots,L_n$ also satisfies $l(A) > l(L_i)$ for $i = 1,\dots,n$. Since I and J agree on all ground atoms of level less than α, we have $J \models L_1 \wedge \dots \wedge L_n$ and hence $A \in T_P(J)$ as required. □

According to the theorem of Priess-Crampe and Ribenboim every strictly contracting function on a spherically complete generalized ultrametric space has a unique fixed point. By Propositions 4.5 and 4.6, the space I_P of all interpretations has been cast into a generalized ultrametric space satisfying the conditions of the Priess-Crampe and Ribenboim theorem relative to T_P. Therefore, application of this theorem yields the existence of a unique fixed point of T_P, that is, a unique supported model of P. We note, however, that the knowledge that P has a unique supported model I has been used in the construction of ϱ, so that the observations just made are really only of theoretical interest.

5 Localizing Unique Supported Model Classes

When studying various classes of programs, the question naturally arises as to how such classes relate to other classes known in the literature. From the definition, it follows immediately that the unique supported model class of all locally hierarchical programs [6] is contained in the class of all locally stratified programs [21]. In this section, we will relate unique supported model classes larger than the locally hierarchical programs to the notion of weak stratification.

It was pointed out in [5, Remark 5.3] that the original definition of weakly stratified programs in [20] is ambiguous since the two conditions

(a) All strata of a program P consist of trivial components only.

(b) All layers of a program P are definite programs.

which were originally used for defining weakly stratified programs are not equivalent. We will call a program *weakly stratified-a* if condition (a) holds, and *weakly stratified-b* if condition (b) holds. For a discussion of this, see [5, Section 5], and we refer to the same publication for notation concerning weakly stratified programs.

In [19], it was shown that each acceptable program [2] is weakly stratified-a. From [11, Corollary 4.3], we immediately obtain that each Φ-accessible program has a total well-founded model, ie. is *effectively stratified* [5]. Again from [5, Proposition 5.4], we obtain that a program which is weakly stratified-b, is also effectively stratified.

It is easy to see that a program which is weakly stratified-b, is also weakly stratified-a. In the opposite direction, we have the following result.

Theorem 5.1 *If P is weakly stratified-a and if there does not exist a clause $A \leftarrow \texttt{body}$ in ground(P) with $\neg A$ occurring in $\texttt{body}$, then P is weakly stratified-b.*

Proof. Since P is weakly stratified-a, all minimal components are trivial. Let $A \leftarrow \texttt{body}$ be a clause in a bottom layer. Without loss of generality assume that $\texttt{body}$ contains some negative literal $\neg B$, ie.[6] $B < A$, with $A \neq B$ by assumption. Since the component containing A is trivial, we obtain $A \not> B$ and therefore we obtain a contradiction. □

It is clear from the last result that a locally hierarchical program is weakly stratified-a if and only if it is weakly stratified-b. This does in fact also hold for locally stratified programs [21].

We will now generalize the result mentioned earlier from [19].

Theorem 5.2 *If P is Φ-accessible, then P is weakly stratified-a and the unique supported model M_P of P is also its weakly perfect-a model.*

[6] "$<$" denotes the dependency relation taking from the dependency graph of P [20].

Proof. Let M_P be the unique supported model of P and let l be its canonical level mapping with respect to Φ. We can also assume without loss of generality that for each level α there exists some $A \in B_P$ with $l(A) = \alpha$.
(1) We first show that all components of the bottom stratum $S(P)$ of P are trivial. Assume that this is not the case, that is, that there exists a minimal component $C \subseteq S(P)$ which is not trivial. Then there must be some $A \in C$ with $l(A)$ minimal, and some $A' \in C$ with $A \neq A'$. Note that $A < A'$ and $A' < A$ [5, Definition 5.1]. Let B be an arbitrary atom occurring in a ground clause with head A. Then $B \leq A$ and $A < A'$, giving $B < A'$, and therefore $B < A$ since $A' < A$. Thus, by minimality of C we obtain $B \in C$. So all atoms B occurring in bodies of clauses in ground(P) with head A belong to C. Since P is Φ-accessible, however, there must exist some choice of B for which we have $l(B) < l(A)$, and this contradicts the minimality of $l(A)$. Note that the bottom stratum contains all atoms of level 0, and hence is non-empty.
(2) The model M of the bottom layer is compatible with M_P, that is, if a literal is true, respectively false, in M, then it is true, respectively false, in M_P. In order to see this, note that for every atom A in a minimal component, the bottom layer $L(P)$ contains all clauses with head A and all clauses with head being any of the body atoms of clauses in the bottom layer. Since the program P is Φ-accessible, it is easy to see that the subprogram formed by the bottom layer is also Φ-accessible and has a unique supported model which is compatible with M_P.
Now let A be an atom in $L(P)$ which occurs negatively in the body of some clause. Since all components are trivial, A must also be the head of the same clause, i.e we have $A < A$. If B is another body atom in the same clause, then we obtain $B < A$ and $A < B$ which contradicts triviality of all components. Hence, if some atom A occurs negatively in a clause in $L(P)$, then the clause is of the form $A \leftarrow \neg A$. All models of $L(P)$ must therefore assign the truth value true to all atoms occurring negatively in $L(P)$. The program which is obtained from omitting all these clauses is definite and has a least model which agrees with M_P. If we add to this model all atoms which occur negatively in $L(P)$, we obtain the least model of $L(P)$.
(3) We show that P/M is Φ-accessible (see [5]). This is indeed the case since (2) holds, and is easily seen by applying Theorem 3.3.
(4) We can now apply steps (1), (2) and (3) via transfinite induction as in [20], which yields that P is indeed weakly stratified-a and that M_P is the weakly perfect-a model of P. Thus, the proof is complete. □

Since locally stratified programs are a generalization of locally hierarchical programs [21], it is clear that each locally hierarchical program has a unique perfect model. This does not hold, however, for Φ^*-accessible programs. Indeed, the program

$$p \leftarrow \neg q$$
$$q \leftarrow r, \neg p$$

is Φ^*-accessible (even acceptable) with respect to the unique supported model $M = \{p\}$. However, $I = \{q\}$ is also a model of this program and while I is preferable to M, M in turn is also preferable to I, so P does not have a perfect model.

Theorem 5.3 *Let P be Φ-accessible. Then P has a unique supported model M_P which is the unique stable model, the well-founded model, a minimal two-valued model, and the weakly perfect-a model of P.*

Proof. We know that $M_P = \Phi_P \uparrow \alpha$ for some ordinal α and that M_P is total. By Theorem 3.6, we know that M_P^+ is a minimal two-valued model of P and, by Theorem 5.2, we know that M_P is the weakly perfect-a model of P. By [11, Corollary 4.3], $M_P = \Phi_P \uparrow \alpha$ is a subset of the well-founded model of P, and since M_P is total, it must coincide with the well-founded model. By [11, Corollary 5.6], totality of the well-founded model implies that it coincides with the unique stable model of the program. This completes the proof. □

A Φ^*-accessible program may not be weakly stratified-b, as can be seen from the following program.

$p \leftarrow$

$p \leftarrow q, \neg p$

The bottom layer contains the clause $p \leftarrow q, \neg p$ and is therefore not a definite program. The program, however, is Φ^*-accessible, even acceptable.

On the other hand, there exist programs $P \in$ USM which are not weakly perfect-a. To see this, note that the following program

$p \leftarrow \neg q$

$q \leftarrow \neg p$

$p \leftarrow \neg p$

has unique supported model $\{p\}$. However, it has $\{p, q\}$ as a minimal component which is not trivial.

6 Unique Supported and Stable Models

The stable model semantics and the supported model semantics share the property that a program may have several meanings under these semantics. Stable models are always supported but not vice versa, so the stable model semantics can be viewed as a refinement of the supported model semantics. In this section, we will discuss some issues relating the two.

Proposition 6.1 *There is a program $P \in USM$ whose well-founded model is not total and which does not have a stable model.*

Proof. Consider the following program P:

$p \leftarrow p$

$p \leftarrow \neg p$

We obtain $T_P(\{p\}) = \{p\}$ and $T_P(\emptyset) = \{p\}$, so $\{p\}$ is the unique supported model of P. However, the Gelfond-Lifschitz transformation using $\{p\}$ deletes the second clause and keeps the first. The resulting program has minimal model $\emptyset$, so $\{p\}$ is not a stable model. Since totality of the well-founded model implies that the well-founded model is stable, we obtain that P does not have a total well-founded model. □

We define well-supported Herbrand models following [8,9].

Definition 6.2 An interpretation I of a program P is called *well-supported* if there exists a strict well-founded partial ordering $\prec$ on I such that for any atom $A \in I$ there exists a clause $A \leftarrow B_1, \dots, B_n, \neg C_1, \dots, \neg C_m$ satisfying $I \models B_1 \wedge \dots \wedge B_n \wedge \neg C_1 \wedge \dots \wedge \neg C_m$ and $B_i \prec A$ for each $i = 1, \dots, n$.

The following theorem was given in [8, Theorem 2.1].

Theorem 6.3 *For a normal logic program P, the well-supported models of P are exactly the stable models of P.*

Given a program P, we will denote by P' the program which is obtained from P as follows: P' is the set of all clauses $A \leftarrow A_1, \dots, A_n$ for which there is a clause $A \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m$ in P. Thus, P' denotes the program which is obtained by omitting all negative literals in all the clauses in P.

We can now characterize a large class of programs for which stable and supported models coincide.

Theorem 6.4 *Let P be a program such that P' is Φ^*-accessible. Then the supported models of P are exactly the stable models of P.*

Proof. Let M be a supported model of P. We show that M is well-supported.
(1) M is a supported model of P/M. In order to show this, let $A \leftarrow \texttt{body}$ be a clause in P/M, and assume that $\texttt{body}$ is true in M. Then the body of a corresponding clause in ground(P) is also true with respect to M by definition of P/M, and hence A is true with respect to M. So M is a model of P/M. To show supportedness, assume that $A \in M$. Then there is a clause $A \leftarrow \texttt{body}$ in P with $M \models \texttt{body}$. By definition of P/M, we obtain that there is a corresponding clause in P/M whose body is true in M. So M is supported as a model of P/M.
(2) Since P' is Φ^*-accessible, it has a unique supported model K. We show that $M \subseteq K$. Assume that this is not the case, so that there is $A \in M \setminus K$ with $l(A)$ minimal. Since M is a supported model of P/M, we know that there is a clause $A \leftarrow \texttt{body}$ in P/M with $M \models \texttt{body}$. But $\texttt{body}$ is also the body of a clause in P' with head A. So, by Φ^*-accessibility of P', and since $A \notin K$ by assumption, there exists a literal B in $\texttt{body}$ with $l(B) < l(A)$ and $K \not\models B$. Since P' is definite we now obtain that $B \in M$ and $B \notin K$ contradicting minimality of $l(A)$ in our choice of A. Thus, $M \subseteq K$.
(3) We show finally that M is well-supported as a model of P. Let $A \in M$. Since M is a supported model of P, there exists a clause $A \leftarrow$

$B_1, \ldots, B_n, \neg C_1, \ldots, \neg C_m$ in ground(P) such that the body of this clause is true in M. From the inclusion $M \subseteq K$ it follows that $B_1, \ldots, B_n \in K$. Now, since P' is Φ^*-accessible we obtain $l(A) > l(B_i)$ for all $i = 1, \ldots, n$. Therefore, the strict ordering $\prec$ on M defined by $B \prec C$ if and only if $l(B) < l(C)$ establishes that the model M is well-supported. □

The result in Theorem 6.4 cannot be generalized by replacing Φ^* with Φ: there exists a program P such that P' is Φ-accessible and such that P has a supported model which is not a stable model. In order to see this, let P be the program given in the proof of Proposition 6.1. Then P' has a unique supported model $M = \{p\}$ and is Φ-accessible. So M is indeed a supported model of P but not a stable model of P.

7 Conclusions and Further Work

The results presented here can be thought of as a study concerned with the "space" of all logic programs: many classes of programs have been described and analysed in the literature, but the relationships between them are not yet satisfactorily understood. In this paper, we have focussed on unique supported model classes, which have very strong properties, and which can perhaps be considered basic classes of programs since many other known classes of programs are larger than these.

Of the various classes described in [14], the Φ- and Φ^*-accessible programs are perhaps the most remarkable from a semantical point of view. Both are natural generalizations of the locally hierarchical and acceptable programs and both classes are computationally adequate while maintaining an unambiguous semantics. The smaller class of Φ^*-accessible programs is perhaps the more natural common generalization of the locally hierarchical and acceptable programs, since it treats all clauses equally, in the spirit of these two classes [14].

The supported model semantics seems to be satisfactory for Φ-accessible programs but not for all programs in the larger class USM. Indeed, programs in USM $\setminus [\Phi]$ seem to be somewhat unnatural from a logic programming point of view since they contain mutually recursive atoms whose truth values cannot be determined from the remaining program. Such a feature might make sense in the context of answer set programming [17], but it remains to be seen to what extent this is the case.

At this stage, it is natural to raise questions about other classes of programs which have unique (or unique total) models under other kinds of semantics, such as the perfect [21], the well-founded, or the stable model semantics, and the authors are attempting to modify the approach presented here and in [14] for these purposes.

References

[1] Apt, K., Marek, V., Truszczyński, M. and Warren, D., editors, "The Logic Programming Paradigm: A 25-Year Perspective," Springer, Berlin, 1999.

[2] Apt, K. and Pedreschi, D., *Reasoning about termination of pure prolog programs*, Information and Computation **106** (1993), pp. 109–157.

[3] Batarekh, A., "Topological Aspects of Logic Programming," Ph.D. thesis, Syracuse University (1989).

[4] Batarekh, A. and Subrahmanian, V., *Topological model set deformations in logic programming*, Fundamenta Informaticae **12** (1989), pp. 357–400.

[5] Bidoit, N. and Froideveaux, C., *Negation by default and unstratifiable logic programs*, Theoretical Computer Science **78** (1991), pp. 85–112.

[6] Cavedon, L., *Continuity, consistency, and completeness properties for logic programs*, in: G. Levi and M. Martelli, editors, *Proceedings of the 6th International Conference on Logic Programming* (1989), pp. 571–584.

[7] Clark, K., *Negation as failure*, in: H. Gallaire and J. Minker, editors, *Logic and Data Bases*, Plenum Press, New York, 1978 pp. 293–322.

[8] Fages, F., *A new fixpoint semantics for general logic programs compared with the well-founded and the stable model semantics*, New Generation Computing **9** (1991), pp. 425–443.

[9] Fages, F., *Consistency of Clark's completion and existence of stable models*, Journal of Methods of Logic in Computer Science **1** (1994), pp. 51–60.

[10] Fitting, M., *A Kripke-Kleene-semantics for general logic programs*, Journal of Logic Programming **2** (1985), pp. 295–312.

[11] Gelder, A. V., Ross, K. and Schlipf, J., *The well-founded semantics for general logic programs*, Journal of the ACM **38** (1991), pp. 620–650.

[12] Gelfond, M. and Lifschitz, V. *The stable model semantics for logic programming*, in: R. Kowalski and K. Bowen, editors, *Logic Programming. Proceedings of the 5th International Conference and Symposium on Logic Programming* (1988), pp. 1070–1080.

[13] Hitzler, P. and Seda, A.K. *Acceptable programs revisited*, in: *Proc. Workshop on Verification in Logic Programming, 16th Int. Conf. on Logic Programming (ICLP'99), Las Cruces, New Mexico*, Electronic Notes in Theoretical Computer Science **30 (1)** (1999), pp. 1–18.

[14] Hitzler, P. and Seda, A.K. *Characterizations of classes of programs by three-valued operators*, in: M. Gelfond, N. Leone and G. Pfeifer, editors, *Logic Programming and Nonmonotonic Reasoning, Proceedings of the 5th International Conference on Logic Programming and Non-Monotonic Reasoning (LPNMR'99), El Paso, Texas, USA*, Lecture Notes in Artificial Intelligence **1730** (1999), pp. 357–371.

[15] Hitzler, P. and Seda, A.K. *The fixed-point theorems of Priess-Crampe and Ribenboim in logic programming*, in: *Valuation Theory and its Applications, Proceedings of the 1999 Valuation Theory Conference, University of Saskatchewan in Saskatoon, Canada*, Fields Institute Communications Series (1999), 17 pages, to appear.

[16] Lloyd, J., "Foundations of Logic Programming," Springer, Berlin, 1988.

[17] Marek, V. and Truszczyński, M., *Stable models and an alternative logic programming paradigm*, in: K. Apt, V. Marek, M. Truszczyński and D. Warren, editors, *The Logic Programming Paradigm: A 25 Year Persepective*, Springer, Berlin, 1999 pp. 375–398.

[18] Prieß-Crampe, S. and Ribenboim, P., *Ultrametric spaces and logic programming*, Journal of Logic Programming **42** (2000), pp. 59–70.

[19] Protti, F. and Zaverucha, G., *On the relations between acceptable programs and stratifiable classes*, in: F. D. Oliveira, editor, *Advances in Artificial Intelligence, 14th Brazilian Symposium on Artificial Intelligence, SBIA '98, Porto Alegre, Brazil*, Lecture Notes in Computer Science **1515** (1998), pp. 141–150.

[20] Przymusinska, H. and Przymusinski, T., *Weakly stratified logic programs*, Fundamenta Informaticae **13** (1990), pp. 51–65, k.R. Apt, editor, special issue of Fundamenta Informaticae on Logical Foundations of Artificial Intelligence.

[21] Przymusinski, T., *On the declarative semantics of deductive databases and logic programs*, in: J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, Morgan Kaufmann, Los Altos, CA, 1988 pp. 193–216.

[22] Przymusinski, T., *On the declarative and procedural semantics of logic programs*, Journal of Automated Reasoning **5** (1989), pp. 167–205.

[23] Seda, A.K., *Topology and the semantics of logic programs*, Fundamenta Informaticae **24** (1995), pp. 359–386.

[24] Willard, S., "General Topology," Addison-Wesley, Reading, MA, 1970.