Strictly Level-Decreasing Logic Programs

Anthony Karel Seda
Department of Mathematics, National University of Ireland
Cork, Ireland
E-mail: aks@bureau.ucc.ie

Pascal Hitzler
Mathematische Fakultät, Universität Tübingen
Tübingen, Germany
E-mail: pascal.hitzler@student.uni-tuebingen.de

Abstract

We study strictly level-decreasing logic programs (sld-programs) as defined earlier by the present authors. It will be seen that sld-programs, unlike most other classes of logic programs, have both a highly intuitive declarative semantics, given as a unique supported model, and are computationally adequate in the sense that every partial recursive function can be represented by some sld-program P. Allowing for a safe use of cuts, an interpreter based on SLDNF-resolution, as implemented for example in standard Prolog systems, is shown to be sound and complete with respect to this class of programs. Furthermore, we study connections between topological dynamics and logic programming which are suggested by our approach to the declarative semantics of sld-programs.

1 Introduction

A programming paradigm consists of a *syntax* or *formal language* and an *interpreter* which assigns a *procedural semantics* to any program satisfying the given syntactical conditions. The procedural semantics of a given program therefore assigns to any allowed input value one or more output values.

Logic programming is distinguished from other programming paradigms by describing the syntax of a logic program P as a set of clauses from first (or higher) order logic, satisfying some additional properties. Viewing P as a set of axioms, a *declarative semantics* for P is given by a distinguished model M for P. Ideally, M represents the meaning intended by the programmer implementing P.

The classical example of this is provided by *definite* or *positive* logic programs. In this case, the syntax is simply the Horn-clause subset of first order predicate logic together with SLD-resolution as interpreter, and the declarative semantics is taken to be the least Herbrand model for the given program.

Horn-clauses, however, do not allow negation symbols in their bodies and therefore lack expressiveness. To overcome this, one usually considers *normal* logic programs, that is to say, programs which consist of a (possibly infinite) set of clauses of the form $A \leftarrow A_1, \dots, A_{k_1}, \neg B_1, \dots, \neg B_{l_1}$ with $k_1, l_1 \geq 0$, where A_i and B_j are atoms. For this class of programs the usual interpreter is SLDNF-resolution, see [5].

It turns out to be difficult to assign a declarative semantics to normal logic programs in full generality. Therefore, various subclasses of logic programs have been proposed in attempts to resolve this issue. These subclasses are often defined by conditions on their syntax, and they vary in relation to the difficulty of assigning them a satisfactory declarative semantics. Indeed, it will be convenient next to briefly review certain of these conditions and relate them to the main topic of this paper. In order to do this, we recall the standard notation B_P for the Herbrand base of a logic program P and the notation $\text{ground}(P)$ for the set of all ground instances of clauses in P, see [5, 11].

1.1 Definition A normal logic program P is called (1) *acyclic*, (2) *strictly level-decreasing (sld)*, (3) *locally stratified*, respectively, if there exists some countable ordinal γ and a mapping $l : B_P \to \gamma$, called a *level mapping*, such that for every clause $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ in $\text{ground}(P)$ and for all i, j we have (1) $\gamma = \omega$ and $l(A_i), l(B_j) < l(A)$, (2) $l(A_i), l(B_j) < l(A)$, (3) $l(A_i) \leq l(A)$ and $l(B_j) < l(A)$, respectively.

It is clear that every acyclic program is strictly level-decreasing and that every strictly level-decreasing program is locally stratified. In fact, the terminology "semi-strictly level-decreasing" was used in place of "locally stratified" in [11], but will not be used here.

Every acyclic program P has a unique supported model M (see [3]). Moreover, it was shown in [1] that SLDNF-resolution as interpreter terminates on all input values. So for acyclic programs, declarative and procedural semantics agree. But, since SLDNF-resolution always terminates, it is not possible to represent every partial recursive function as an acyclic program. The class of all acyclic programs is therefore not computationally adequate.

On the other hand, locally stratified programs do not have unique supported models, but each such program does have a unique *perfect* model as defined and constructed in [8]. Furthermore, the class of all locally stratified programs contains the class of all definite programs and is therefore computationally adequate since the latter class has this property, see [13]. To elaborate a little further on the issues involved, let P be the following (locally stratified) program

$$p(0) \leftarrow \neg q(0)$$
$$q(0) \leftarrow q(0)$$

Here, we have two minimal and supported models $\{p(0)\}$ and $\{q(0)\}$ for P, and it is natural to try to find a way of showing that one is preferable to the other in some sense. In [8] it is argued that the perfect model provides such a way, and in fact for the program in question the perfect model semantics coincides with the first model, $\{p(0)\}$, which may therefore be viewed as preferable to the second model, $\{q(0)\}$. On the other hand, both are models of the Clark-completion $\text{comp}(P)$ of P, due to the second clause in the progam P. It could in fact be argued that if a programmer explicitly implements this second clause, he intends to allow $\{q(0)\}$ as a model for the program, which is consistent with the view of $\text{comp}(P)$ as a set of first-order clauses. The perfect model semantics ignores this second clause and assumes that the programmer did not intend the logical meaning underlying it. These points hint at some of the problems involved in finding a satisfactory semantics for locally stratified programs. From the procedural point of view, a practically implementable interpreter which can deal with the power of expressiveness of locally stratified programs remains to be found. Nevertheless, it is safe to say that locally stratified programs form one of the most important classes of programs in AI and in the study of disjunctive databases.

The class of all strictly level-decreasing programs, however, does not have the undesirable properties of the other two classes of programs just discussed. Indeed, it was shown in [11] that every sld-program has a unique supported model, i.e. the completion $\text{comp}(P)$ of P has a unique model. Moreover, it was shown how this model can be constructed by a transfinite sequence. It will be shown here that the class of sld-programs is computationally adequate, and indeed the structure of the paper is as follows.

Section 2 of the present paper is devoted to the declarative and procedural semantics of sld-programs. We briefly review the results already obtained in [11] and investigate in Section 2.1 simple methods to construct the unique supported model. In Section 2.2, we will allow the use of safe cuts in the syntax of sld-programs and show that these together with an interpreter based on SLDNF-resolution, as implemented in standard Prolog systems, can represent every partial recursive function.

Our approach to the declarative semantics of sld-programs employs methods which suggest connections between logic programming and topological dynamics. In Section 3, we therefore investigate these and connect them up with our previous observations on this topic presented in [11]. In particular, in Sections 3.1 and 3.2 we study the Vietoris space of B_P and iterated function systems in the context of arbitrary normal logic programs. These topics, i.e. Vietoris space and iterated function systems, have both been brought into prominence by the work of Abbas Edalat, see [2]. Finally, we conclude with some further observations on sld-programs which relate them yet more closely to topological dynamics. Our overall claim is that the class of strictly level-decreasing programs is particularly interesting, both computationally and mathematically, and the objective of this paper is to substantiate this claim.

2 Semantics of Strictly Level-Decreasing Logic Programs

It will be convenient to recall first some basic notions used in the sequel. Details and further background can be found in [5].

Let P denote a normal logic program with underlying first order language $\mathcal{L}$. As usual, U_P and B_P will denote respectively the Herbrand universe and Herbrand base of P, and I_P or $\mathcal{P}(B_P)$ will denote the set of all Herbrand interpretations for $\mathcal{L}$ (or for P). We let $T_P : I_P \to I_P$ denote the usual immediate consequence operator associated with P. Throughout, l will denote a level mapping; thus l is simply a mapping $l : B \to \gamma$, where B is a set and γ is an arbitrary countable ordinal (usually, B will be the Herbrand base B_P). For any such mapping, we set $\mathcal{L}_\alpha = \{a \in B; l(a) < \alpha\}$ and let Γ_γ be the set of symbols $\{2^{-\alpha}; \alpha < \gamma\}$ ordered by $2^{-\alpha} < 2^{-\beta}$ if and only if $\beta < \alpha$. (In the case $B = B_P$, we will denote a typical element of B_P, i.e. a ground atom, by A rather than by the lower case a).

Given the set B, the *Cantor topology* Q on $D = 2^B$ is characterized via convergence as follows: a net I_λ converges in Q if and only if every $a \in B$ is either eventually in I_λ or eventually not in I_λ. In particular, this is so when B is countable (for example when $B = B_P$), in which case sequences suffice to describe topological concepts and the stated convergence criterion still applies. This fact, which we will refer to as the *convergence criterion*, will be used in several places and is recorded as Proposition 4 in [9]. Finally, for a net I_λ, let $\mathrm{gl}(I_\lambda)$ denote the greatest limit of I_λ in the Scott topology on D, see [9]. It was shown in [10] that if $I_\lambda \to I$ in Q, then $I = \mathrm{gl}(I_\lambda)$ and that $\mathrm{gl}(I_\lambda) = \{a \in B; a \in I_\lambda \text{ eventually}\}$. The Cantor topology and its rôle in logic programming was studied in [9, 10] and in [11, 12].

In [11], it was shown that every sld-program P has a unique supported model M_P which can therefore be understood to be the standard semantics for P. In order to make this paper relatively self-contained, we briefly review the main points of our construction of M_P in [11], and refer the reader to that paper for details.

Let P denote a normal logic program which is locally stratified with respect to a level mapping $l : B_P \to \gamma$, where $\gamma \geq 1$. For each n satisfying $0 < n \leq \gamma$, let $P_{[n]}$ denote the set of all clauses in ground(P) in which only atoms A with $l(A) < n$ occur. We define $T_{[n]} : \mathcal{P}(\mathcal{L}_n) \to \mathcal{P}(\mathcal{L}_n)$ by $T_{[n]}(I) = T_{P_{[n]}}(I)$. Next, we construct the transfinite sequence $(I_n)_{n\in\gamma}$ inductively as follows. For each $m \in N$, we put $I_{[1,m]} = T_{[1]}^m(\emptyset)$ and set $I_1 = \bigcup_{m=0}^\infty I_{[1,m]}$. If $n \in \gamma$, where $n > 1$ is a successor ordinal, then for each $m \in N$ we put $I_{[n,m]} = T_{[n]}^m(I_{n-1})$ and set $I_n = \bigcup_{m=0}^\infty I_{[n,m]}$. If $n \in \gamma$ is a limit ordinal, we put $I_n = \bigcup_{m<n} I_m$. Finally, we put $I_{[P]} = M_P = \bigcup_{n<\gamma} I_n$.

Next, we recall a construction related to domain theory which we made in [11]. Let B be a countable

set and let $l : B \to \gamma$ be a level mapping. It is well-known that $D = 2^B$ is a domain with respect to set inclusion. The level mapping l induces a distance function $d_l = d : D \times D \to \Gamma_{\gamma+1}$ via $d(I, J) = \min\{2^{-\alpha}; a \in I$ if and only if $a \in J$ for all $a \in B$ with $l(a) < \alpha\}$. This distance function is in fact a *generalized ultrametric* as defined in [6], i.e. it satisfies the following conditions, where we denote $2^{-\gamma}$ by 0:
(1) $d(I, I) = 0$ for all $I \in D$.
(2) $d(I, J) = d(J, I)$ for all $I, J \in D$.
(3) If $d(I, J) \leq \alpha$ and $d(J, K) \leq \alpha$, then $d(I, K) \leq \alpha$.

Now let P be a program which is strictly level-decreasing with respect to some level mapping l and take $B = B_P$. Then l induces a generalized ultrametric on I_P, and T_P is *strictly contracting*, i.e. it satisfies $d(T_P(I), T_P(J)) < d(I, J)$ for all $I, J \in I_P$. Moreover, I_P and T_P satisfy the hypotheses of the theorem of Priess-Crampe and Ribenboim (see [6, 7]) which therefore yields the existence of a unique fixed point of T_P and therefore of a unique supported model for P.

2.1 Declarative Semantics

We begin our study of sld-programs by showing how such a program P can be endowed with a canonical level mapping l_P which is smallest in a certain obvious sense.

2.1 Definition Let P be a program which is strictly level-decreasing with respect to a level mapping l. We define a level mapping l_P on B_P as follows. For every $A \in B_P$ which does not occur as a head in ground(P), let $l_P(A) = 0$. For every $A \in B_P$ which occurs as the head of a unit clause but not as the head of any non-unit clause, let $l_P(A) = 0$. Now let $A \in B_P$ be such that A is the head of some clause(s) in ground(P). Let $\mathcal{B}_A$ be the collection of body-literals occurring in these clauses. Note that $\mathcal{B}_A$ is finite for every A if P has no local variables. Now suppose that for every $B \in \mathcal{B}_A$, $l_P(B)$ is already defined. Let $M_A = \sup_{B \in \mathcal{B}_A} l_P(B)$ and set $l_P(A) = M_A + 1$, if M_A is a successor ordinal, and set $l_P(A) = M_A$, if M_A is a limit ordinal. Then l_P is obtained by transfinitely iterating this procedure. We will refer to l_P, as defined above, as the (*canonical*) *level mapping* of P and, further, γ_P will denote the smallest ordinal α such that $l_P(A) \in \alpha$ for all $A \in B_P$.

2.2 Proposition Let P be a program which is strictly level-decreasing with respect to some level mapping l. Then l_P, as defined above, is a function on B_P and P is strictly level-decreasing with respect to l_P. Moreover, if P has no local variables, then $\gamma_P \leq \omega$ and hence P is acyclic[1] and acceptable[1].

Proof: First we show that dom$(l_P) = B_P$. Suppose there is $A \in B_P \setminus$ dom(l_P); we can further suppose that $l(A)$ is minimal for A with this property. Then there must be some $B \in \mathcal{B}_A$ with $B \notin$ dom(l_P), otherwise $l_P(A)$ is defined in the process given in Definition 2.1. Since P is an sld-program, we have $l(B) < l(A)$ which contradicts the choice of A with $l(A)$ minimal. Therefore, l_P is a level mapping, and obviously P is strictly level-decreasing with respect to it. Finally, if P has no local variables, then the set $\mathcal{B}_A$ is finite for every $A \in B_P$, and so l_P maps into ω. Hence, $\gamma_P \leq \omega$. ■

The construction above of the level mapping l_P can be used to determine whether or not a given program P is strictly level-decreasing, and the following corollary is immediate.

[1] See [1].

2.3 Corollary Let P be an arbitrary normal logic program. Then P is strictly level-decreasing if and only if $\mathrm{dom}(l_P) = B_P$.

2.4 Proposition Let P be a program which is strictly level-decreasing with respect to a level mapping l. Then for every $A \in B_P$, we have $l_P(A) \leq l(A)$.

Proof: Suppose the conclusion is false. Thus, there is $A \in B_P$ with $l(A) < l_P(A)$, and such that $l(A)$ is minimal. Then, for all $B \in \mathcal{B}_A$, we have $l(B) < l(A)$ because P is strictly level-decreasing. Therefore, by minimality of $l(A)$, we have $l(B) \geq l_P(B)$ for all $B \in \mathcal{B}_A$. By definition of l_P, we see that $l_P(A) = \min\{\alpha; \alpha > l_P(B), B \in \mathcal{B}_A\} \leq \min\{\alpha; \alpha > l(B), B \in \mathcal{B}_A\} \leq l(A)$. From this we obtain $l_P(A) \leq l(A)$, giving the required contradiction. ■

As stated in the introduction to this section, a procedure for finding the perfect model M_P for locally stratified programs P was given by us in [11]. Our approach shed light on the original construction made by Przymusinski in [8], and involved studying convergence in Q of certain transfinite sequences of iterates which were carefully controlled in relation to the level mapping associated with the local stratification. One particular fact which emerged from that study, see [11, Theorem 5], was that for acyclic programs P, M_P can be realised as the limit in Q of the sequence $T_P^n(I)$ for any choice of $I \in I_P$ (in particular, the choice of $I = \emptyset$ is especially natural). The question arises as to whether or not this result can be generalized to programs which are strictly level-decreasing relative to an arbitrary level mapping. The following example shows that the answer to this question is negative.

2.5 Example Consider the following program P:

$$\begin{aligned} t(0) &\leftarrow p(X) \\ p(s(X)) &\leftarrow r(X) \\ r(s(X)) &\leftarrow p(X) \\ p(0) &\leftarrow \neg q(0) \\ q(0) &\leftarrow \end{aligned}$$

and define the level mapping l on B_P by: $q(s^n(0)) \mapsto 0, p(s^n(0)) \mapsto n+1, r(s^n(0)) \mapsto n+1$, and $t(s^n(0)) \mapsto \omega$. Then it is clear that P is strictly level-decreasing with respect to l. However, on computing the elements of the sequence $T_P^n(\emptyset)$, we find that $t(0)$ belongs to infinitely many of the $T_P^n(\emptyset)$ and does not belong to infinitely many others. Therefore, by the convergence criterion, the sequence $T_P^n(\emptyset)$ does not converge in Q and we see that M_P cannot be computed by simply iterating T_P, by contrast with the case of ω-level mappings.

Nevertheless, considerable simplification does arise in constructing M_P for sld-programs as distinct from general locally stratified programs, and the remaining results of this section demonstrate this fact; much of this simplification ultimately rests on the recursion equations in [11, Corollary 2]. For the rest of this section, unless otherwise stated, P will denote an arbitrary sld-program with level mapping l, and T will denote T_P.

2.6 Lemma Suppose that P is a locally stratified normal logic program. Then for any ordinal $n \geq 1$ and subset $J \subseteq B_P$, we have:
(1) $T_{[n]}(J \cap \mathcal{L}_n) = T_{[n]}(J)$, and

(2) $T_{P(n)}(J \cap \mathcal{L}_{n+1}) = T_{P(n)}(J)$.
If, further, P is strictly level-decreasing, then we also have:
(3) $T_{[n+1]}(J \cap \mathcal{L}_n) = T_{[n+1]}(J)$, and
(4) $T_{P(n)}(J \cap \mathcal{L}_n) = T_{P(n)}(J)$.

Proof: (1) Suppose that $A \in T_{[n]}(J \cap \mathcal{L}_n)$. Then there is a clause in $P_{[n]}$ of the form $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ such that $A_1, \ldots, A_{k_1} \in J \cap \mathcal{L}_n$ and $B_1, \ldots, B_{l_1} \notin J \cap \mathcal{L}_n$. Since $l(A) < n$, we have $l(A_i), l(B_j) < n$ for all i, j and so $A_1, \ldots, A_{k_1}, B_1, \ldots, B_{l_1} \in \mathcal{L}_n$. Therefore, $A_1, \ldots, A_{k_1} \in J$ and $B_1, \ldots, B_{l_1} \notin J$ from which we obtain that $A \in T_{[n]}(J)$.

Conversely, suppose that $A \in T_{[n]}(J)$. Then there is a clause in $P_{[n]}$ of the form $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ such that $A_1, \ldots, A_{k_1} \in J$ and $B_1, \ldots, B_{l_1} \notin J$. By level considerations again, we have $A_1, \ldots, A_{k_1} \in J \cap \mathcal{L}_n$ and $B_1, \ldots, B_{l_1} \notin J \cap \mathcal{L}_n$. Therefore, $A \in T_{[n]}(J \cap \mathcal{L}_n)$.

The remaining statements are proved similarly taking note of the appropriate equalities and inequalities between the values $l(A)$ and $l(A_i), l(B_j)$ for the atoms occurring in a typical clause $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ in $P_{[n]}$ and in $P(n)$. ■

It will be convenient to recall next a definition made in [4, Definition 7.11] for a normal logic program P. Again writing T for T_P, we define $T^{(n)} : \mathcal{P}(\mathcal{L}_n) \to \mathcal{P}(\mathcal{L}_n)$ by $T^{(n)}(I) = T(I) \cap \mathcal{L}_n$, where n is an arbitrary ordinal satisfying $1 \leq n \leq \gamma$. In fact, this statement differs slightly from that given in [4] to the extent that $\mathcal{L}_n$ as used here differs slightly from its meaning in [4].

2.7 Lemma Suppose that P is a locally stratified normal logic program. Then the following statements hold.
(1) For any subset $I \subseteq B_P$ and any ordinal satisfying $1 \leq n \leq \gamma$, we have $T_{[n]}(I) = T^{(n)}(I)$ $(= T(I) \cap \mathcal{L}_n)$.
(2) If, further, P is strictly level-decreasing, then for any ordinal n satisfying $1 \leq n \leq \gamma$, we have $I_{n+1} = I_n \cup T_{P(n)}(I_n) = T_{[n+1]}(I_n)$.

Proof: Statement (1) is simply a reiteration of [4, Proposition 7.17] allowing for the difference, already mentioned, between the meaning of $\mathcal{L}_n$ as used here and in [4].

For (2), we have $I_{n+1} = \bigcup_{m=0}^{\infty} I_{[n+1,m]} = \bigcup_{m=0}^{\infty} I_n \cup T_{P(n)}(I_n)$ by [11, Corollary 2]. Therefore, $I_{n+1} = I_n \cup T_{P(n)}(I_n) = T_{[n]}(I_n) \cup T_{P(n)}(I_n) = T_{[n+1]}(I_n)$, using the fact that I_n is a fixed point of $T_{[n]}$. ■

2.8 Theorem Suppose that P is a strictly level-decreasing normal logic program. Then for every limit ordinal α and $k \in \mathbb{N}$ we have $T^k(I_\alpha) \cap \mathcal{L}_{\alpha+k} = I_{\alpha+k}$, and for $\beta = \alpha + \omega$ we have $\lim_{n \in \mathbb{N}} (T_{[\beta]})^n (I_\alpha) = I_\beta$ in Q.

Proof: The first statement follows from Lemmas 2.6 and 2.7 by induction. Indeed, for $k = 0$, the statement obviously holds on noting that $I_\alpha \subseteq \mathcal{L}_\alpha$. Now suppose the statement holds for some $k \geq 0$. Then by Lemma 2.7 (2), the induction hypothesis, Lemma 2.6 (1) and (4), and Lemma 2.7 (1) applied in that order we have $I_{\alpha+k+1} = T_{[\alpha+k+1]}(I_{\alpha+k}) = T_{[\alpha+k+1]}(T^k(I_\alpha) \cap \mathcal{L}_{\alpha+k}) = T_{[\alpha+k]}(T^k(I_\alpha) \cap \mathcal{L}_{\alpha+k}) \cup T_{P(\alpha+k)}(T^k(I_\alpha) \cap \mathcal{L}_{\alpha+k})) = T_{[\alpha+k]}(T^k(I_\alpha)) \cup T_{P(\alpha+k)}(T^k(I_\alpha))) = T_{[\alpha+k+1]}(T^k(I_\alpha)) = T(T^k(I_\alpha)) \cap \mathcal{L}_{\alpha+k+1} = T^{k+1}(I_\alpha) \cap \mathcal{L}_{\alpha+k+1}$.

For the second statement, we have $(T_{[\beta]})^n(I_\alpha) \cap \mathcal{L}_{\alpha+n} = I_{\alpha+n}$ for all $n \in \mathbb{N}$ by induction on n. Indeed, the statement clearly holds for $n = 0$. Now suppose it holds for some $n \geq 0$. Then, using Lemma 2.7 (1), Lemma 2.6 (3), the induction hypothesis and Lemma 2.7 (2), we have $(T_{[\beta]})^{n+1}(I_\alpha) \cap$

$\mathcal{L}_{\alpha+n+1} = T_{[\beta]}((T_{[\beta]})^n(I_\alpha)) \cap \mathcal{L}_{\alpha+n+1} = T((T_{[\beta]})^n(I_\alpha)) \cap \mathcal{L}_\beta \cap \mathcal{L}_{\alpha+n+1} = T((T_{[\beta]})^n(I_\alpha)) \cap \mathcal{L}_{\alpha+n+1} = T_{[\alpha+n+1]}((T_{[\beta]})^n(I_\alpha)) = T_{[\alpha+n+1]}((T_{[\beta]})^n(I_\alpha) \cap \mathcal{L}_{\alpha+n}) = T_{[\alpha+n+1]}(I_{\alpha+n}) = I_{\alpha+n+1}$. Now let $A \in B_P$ be arbitrary. Then $l(A) = \alpha + m_0$ for some limit ordinal $\alpha \geq 0$ and some $m_0 \in \mathbb{N}$. Thus, $A \in \mathcal{L}_{\alpha+m_0+1}$ for some α and m_0. Now either $A \in M_P$ or $A \notin M_P$, where $M_P = \bigcup_{n<\gamma} I_n$. If $A \in M_P$, then we must have $A \in I_{\alpha+m_0+1}$ by [11, Lemma 1], otherwise $A \notin M_P$. Hence, $A \in (T_{[\beta]})^{m_0+1}(I_\alpha)$ and therefore $A \in (T_{[\beta]})^m(I_\alpha)$ eventually. If $A \notin M_P$, then A is eventually not in $(T_{[\beta]})^m(I_\alpha)$. Hence, by the convergence criterion, $(T_{[\beta]})^m(I_\alpha)$ converges in Q and obviously converges to I_β. ■

The result just obtained allows us to further simplify the calculation of M_P, and to do this it will be convenient to establish the following terminology.

2.9 Definition Let P be a strictly level decreasing-program. For every limit ordinal $0 \leq \alpha < \gamma_P$ and $\beta = \alpha + \omega$ the next limit ordinal, we define P_β to be the set of all clauses in ground(P) with head A such that $\alpha \leq l(A) < \beta$. The usual order on the limit ordinals induces an order on the set of all P_β. We call each P_β an *ω-component* of P.

2.10 Lemma Suppose that P is a strictly level-decreasing program, let $\alpha < \gamma_P$ be a limit ordinal and let $\beta = \alpha + \omega$. Then for every $n \in \mathbb{N}$, we have $I_\alpha = T_{[\alpha]}((T_{[\beta]})^n(I_\alpha))$.

Proof: We have to show that $d_l(I_\alpha, (T_{[\beta]})^n(I_\alpha)) \leq 2^{-\alpha}$ which is obviously true for $n = 0$. Suppose it holds for some $n \geq 0$. Since $T_{[\beta]} = T_{P_\beta} \cup T_{[\alpha]}$ and since I_α is a fixed point of $T_{[\alpha]}$, we have $d_l(I_\alpha, T_{[\beta]}(I_\alpha)) \leq 2^{-\alpha}$. Since $T_{[\beta]}$ is strictly contracting with respect to d_l, we have $d_l(T_{[\beta]}(I_\alpha), (T_{[\beta]})^{n+1}(I_\alpha)) < d_l(I_\alpha, (T_{[\beta]})^n(I_\alpha))$. By the strong triangle inequality we get $d_l(I_\alpha, (T_{[\beta]})^{(n+1)}(I_\alpha)) \leq \max\{2^{-\alpha}, d_l(I_\alpha, (T_{[\beta]})^n(I_\alpha))\}$ and inductively $d_l(I_\alpha, (T_{[\beta]})^{n+1}(I_\alpha)) \leq d_l(I_\alpha, I_\alpha) \leq 2^{-\alpha}$. ■

We have the following theorem, where we recall from [4, Definition 7.11] that $T \Uparrow n(I)$ is defined inductively for any $I \in I_P$ by setting $T \Uparrow 0(I) = I$ and $T \Uparrow (n+1)(I) = T(T \Uparrow n(I)) \cup I$ for $n \geq 0$.

2.11 Theorem Let P be a strictly level-decreasing program. Then for every limit ordinal $\alpha < \gamma_P$ and $\beta = \alpha + \omega$, we have $I_\beta = \lim_n T_{P_\beta} \Uparrow n(I_\alpha)$ in Q.

Proof: The statement follows from Theorem 2.8 by the observation that

$$T_{P_\beta} \Uparrow n(I_\alpha) = (T_{[\beta]})^n(I_\alpha) \tag{1}$$

for every $n \in \mathbb{N}$, which we will prove by induction on n. Indeed, Equation (1) holds for $n = 0$. So suppose it holds for some $n \in \mathbb{N}$. Then, by the induction hypothesis and the previous lemma, we have $T_{P_\beta} \Uparrow (n+1)(I_\alpha) = T_{P_\beta}(T_{P_\beta} \Uparrow n(I_\alpha)) \cup I_\alpha = T_{P_\beta}((T_{[\beta]})^n(I_\alpha)) \cup I_\alpha = T_{P_\beta}((T_{[\beta]})^n(I_\alpha)) \cup T_{[\alpha]}((T_{[\beta]})^n(I_\alpha)) = T_{[\beta]}((T_{[\beta]})^n(I_\alpha)) = (T_{[\beta]})^{n+1}(I_\alpha)$ which establishes Equation (1) for every $n \in \mathbb{N}$. ■

If we put $I_\beta = \bigcup_{\alpha<\beta} I_\alpha$ for every limit ordinal β which is not of the form used in the theorem, then we obtain a method for calculating M_P. Indeed, we will next give a proof of the fact that every sld-program has a unique supported model. The proof is independent of the results in [11].

Let P be an sld-program. Let $T_\omega : \mathcal{P}(\mathcal{L}_\omega) \to \mathcal{P}(\mathcal{L}_\omega)$ be defined by $T_\omega = T_{P_\omega}$ (= $T_{[\omega]}$). Then, by [11, Theorem 5], T_ω is a contraction with contractivity factor at most $\frac{1}{2}$ with respect to Fitting's complete ultrametric, and therefore has a unique fixed point J_ω. (Fitting in [3] defined a complete ultrametric d

determined by a level mapping $l : B_P \to \omega$ as follows: if $I_1 = I_2$, we set $d(I_1, I_2) = 0$, otherwise we set $d(I_1, I_2) = 2^{-n}$, where I_1 and I_2 differ on some ground atom of level n but agree on all atoms of lower level).

For every limit ordinal $\alpha > 0$ and $\beta = \alpha + \omega$, define inductively $T_\beta : \mathcal{P}(\mathcal{L}_\beta) \to \mathcal{P}(\mathcal{L}_\beta)$ by $T_\beta(I) = T_{P_\beta}(I) \cup J_\alpha$. We show next that T_β is a contraction with contractivity factor at most $\frac{1}{2}$. To do this we define the level mapping $l_\beta : \mathcal{L}_\beta \to \omega$ by $l_\beta(A) = 0$ if $l(A) < \alpha$, and $l_\beta(A) = 1 + (l(A) - \alpha)$ if $\alpha \leq l(A) < \beta$. Now let d_β denote Fitting's ultrametric determined by l_β. Notice that P_β is strictly level-decreasing with respect to l_β but that $P_{[\alpha]}$ does not have this property.

2.12 Lemma With the notation just established, the mapping T_β is a contraction with respect to d_β.

Proof: Suppose that I_1 and I_2 are arbitrary elements of $\mathcal{P}(\mathcal{L}_\beta)$ and that $d_\beta(I_1, I_2) = 2^{-n}$, where $n \geq 0$.
Case 1. $n = 0$.
In this case, I_1, I_2 differ on some element of $\mathcal{L}_\beta$ of l_β-level 0. Let $A \in \mathcal{L}_\beta$ satisfy $l_\beta(A) = 0$, but otherwise be arbitrary; then $A \in \mathcal{L}_\alpha$. If $A \in J_\alpha$, then $A \in T_\beta(I_1)$ and $A \in T_\beta(I_2)$; if $A \notin J_\alpha$, then $A \notin T_\beta(I_1)$ and $A \notin T_\beta(I_2)$. Thus, $T_\beta(I_1)$ and $T_\beta(I_2)$ agree on all atoms of l_β-level 0. Therefore,

$$d_\beta(T_\beta(I_1), T_\beta(I_2)) \leq \frac{1}{2} = \frac{1}{2} d_\beta(I_1, I_2).$$

Case 2. $n > 0$.
In this case, I_1 and I_2 differ on some ground atom of l_β-level n but agree on all ground atoms of lower l_β-level. Let $A \in \mathcal{L}_\beta$ be arbitrary with $l_\beta(A) \leq n$. Suppose further that $A \in T_\beta(I_1)$. If $l_\beta(A) = 0$, then $A \in \mathcal{L}_\alpha$ and hence $A \in J_\alpha$. Therefore, $A \in T_\beta(I_2)$ also. On the other hand, if $l_\beta(A) > 0$, then there is a clause $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ in P_β, with $k_1 \geq 0$ and $l_1 \geq 0$, which satisfies $I_1 \models A_1 \wedge \cdots \wedge A_{k_1} \wedge \neg B_1 \wedge \cdots \wedge \neg B_{l_1}$. If $k_1 = 0 = l_1$, then the clause in question is a unit clause and immediately we have $A \in T_\beta(I_2)$. If $k_1 \neq 0$ or $l_1 \neq 0$ or both, then I_1 and I_2 agree on the atoms $A_1, \ldots, A_{k_1}, B_1, \ldots, B_{l_1}$ since their l_β-levels are lower than n. Therefore, $I_2 \models A_1 \wedge \cdots \wedge A_{k_1} \wedge \neg B_1 \wedge \cdots \wedge \neg B_{l_1}$ and we have $A \in T_\beta(I_2)$. The converse argument holds similarly and so we see that $T_\beta(I_1)$ and $T_\beta(I_2)$ agree on all ground atoms of l_β-level $\leq n$. Therefore,

$$d_\beta(T_\beta(I_1), T_\beta(I_2)) \leq 2^{-(n+1)} = \frac{1}{2} d_\beta(I_1, I_2)$$

as required. ■

It follows from the previous lemma that T_β has a unique fixed point J_β. For every limit ordinal β which is not of the form $\alpha + \omega$ for any limit ordinal α, we set $J_\beta = \bigcup_{\alpha < \beta} J_\alpha$, where the α are limit ordinals. The resulting transfinite sequence (J_α) is increasing since for $\beta = \alpha + \omega$, where α is some limit ordinal, J_β satisfies $J_\beta = T_\beta(J_\beta) = T_{P_\beta}(J_\beta) \cup J_\alpha$, so that $J_\beta \supseteq J_\alpha$. Finally, on taking β to be a limit ordinal such that $\gamma_P \leq \beta$, we obtain $J_\beta = M_P$ i.e. the unique fixed point of T_P.

2.2 Procedural Semantics

For convenience, we establish the following notation for every sld-program P. For $A \in B_P$, we say that $P \models A$ if and only if $A \in M_P$. We say that $P \vdash_{\text{SLDNF}} A$ if and only if there is an SLDNF-derivation for $P \cup \{\leftarrow A\}$.

2.13 Theorem Let P be a strictly level-decreasing program and $A \in B_P$ with $P \vdash_{\text{SLDNF}} A$. Then $P \models A$. If $\gamma_P = \omega$, then $P \vdash_{\text{SLDNF}} A$ if and only if $P \models A$. In particular, if P is without local variables, then $P \models A$ if and only if $P \vdash_{\text{SLDNF}} A$.

Proof: By [5, Proposition 14.2], M_P is the unique model of $\text{comp}(P)$. By [5, Theorem 15.4], the first statement immediately holds. Now let $\gamma_P = \omega$ and $P \models A$. Then, by [1, Corollay 4.11], all SLDNF-derivations of $P \cup \{\leftarrow A\}$ are finite and, therefore, $P \vdash_{\text{SLDNF}} A$ which proves the second statement. If P is without local variables, then P is acyclic by Proposition 2.2, which completes the proof using the second statement. ■

We establish next one of our main results: that every partial recursive function can be computed by an sld-program with cuts. We take the point of view (following [5]) that a cut does not affect the declarative semantics of a program. When talking about SLDNF-resolution for strictly level-decreasing programs with cuts, we assume that the selection function always selects the leftmost literal and, as discussed in [5], that the cut "prunes" the search tree. To obtain a well-defined procedural semantics of a given program, we assume that the topmost clause whose head unifies with a current goal is always selected first, as implemented in standard Prolog systems. So, for what follows, SLDNF-resolution is performed in the way just described.

For convenience, we will denote ground terms by lowercase letters and variables by uppercase letters when refering to a predicate. Thus, $p(x_1, \ldots, x_n, Y)$ means that all x_i are ground and Y is a variable. We write $(P, A) \vdash_{\text{SLDNF}} B$ if $P \cup \{\leftarrow A\}$ has an answer substitution θ (via SLDNF-resolution) such that $A\theta = B$.

2.14 Theorem Identify $\mathbb{N}$ with the set $\{s^n(0); n \in \mathbb{N}\}$ by taking s to be the successor function. Let f be an n-ary partial recursive function. Then there exists a strictly level-decreasing program P_f with cuts and an $(n+1)$-ary predicate symbol p_f such that the following hold:

1. A call to P_f with goal $p_f(x_1, \ldots, x_n, Y)$ or $p_f(x_1, \ldots, x_n, y)$ terminates via SLDNF-resolution if $(x_1, \ldots, x_n) \in \text{dom}(f)$ and backtracking over the goal fails immediately.

2. $(P_f, p_f(x_1, \ldots, x_n, Y)) \vdash_{\text{SLDNF}} p_f(x_1, \ldots, x_n, y)$ if and only if $(x_1, \ldots, x_n) \in \text{dom}(f)$ and $f(x_1, \ldots, x_n) = y$.

3. For every $p_f(x_1, \ldots, x_n, y) \in B_P$ the following are equivalent:

   (a) $P \models p(x_1, \ldots, x_n, y)$

   (b) $P \vdash_{\text{SLDNF}} p(x_1, \ldots, x_n, y)$

   (c) $f(x_1, \ldots, x_n) = y$.

Proof: We follow [13] and [5] with modifications where necessary. The proof is by induction on the number q of applications of composition, primitive recursion, and minimalization needed to define f.

Suppose first that $q = 0$. Thus f must be either the zero function, the successor function, or a projection function.

Zero function

Suppose that f is the zero function defined by $f(x) = 0$. Define P_f to be the program $p_f(X, 0) \leftarrow$.

Successor function

Suppose that f is the successor function defined by $f(x) = s(x)$. Define P_f to be the program $p_f(X, s(X)) \leftarrow$.

Projection function

Suppose that f is the projection function defined by $f(x_1, \ldots, x_n) = x_j$ for some $j \in \{1, \ldots, n\}$. Define P_j to be the program $p_f(X_1, \ldots, X_n, X_j) \leftarrow$.

Clearly, for each of the basic functions, the program P_f, as defined, is an sld-program with the desired properties.

Next, suppose that the partial recursive function f is defined by $q > 0$ applications of composition, primitive recursion, and minimalization.

Composition

Suppose that f is defined by $f(x_1, \ldots, x_n) = h(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$ where $g_1, \ldots, g_m$ and h are partial recursive functions. By the induction hypothesis, corresponding to each g_i (or h), there is an sld-program P_{g_i} (P_h) with cuts and a predicate symbol p_{g_i} (p_h) satisfying the conclusions of the theorem. We can suppose that the programs $P_{g_1}, \ldots, P_{g_m}, P_h$ do not have any predicate symbols in common. Define P_f to be the union of these programs together with the clause

$$p_f(X_1, \ldots, X_n, Z) \leftarrow p_{g_1}(X_1, \ldots, X_n, Y_1), \ldots, p_{g_m}(X_1, \ldots, X_n, Y_m), h(Y_1, \ldots, Y_m, Z), !.$$

Obviously, P_f is an sld-program with cuts. Statement 1 is immediate under the assertion of the induction hypothesis, as is the ‘if’-part of statement 2. The ‘only-if’ part is shown as in [5]. For statement 3, the equivalence of 3a and 3c is immediate and the equivalence of 3b and 3c is shown in a manner analogous to that employed in [13].

Primitive recursion

Suppose that f is defined by

$$\begin{aligned} f(x_1, \ldots, x_n, 0) &= h(x_1, \ldots, x_n) \\ f(x_1, \ldots, x_n, y+1) &= g(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y)) \end{aligned}$$

where h and g are partial recursive functions. By the induction hypothesis, corresponding to h (resp. g), there is an sld-program P_h (resp. P_g) with cuts and a predicate symbol p_h (resp. p_g) satisfying the conclusions of the theorem. We can also suppose that P_h and P_g do not have any predicate symbols in common. Define P_f to be the union of P_h and P_g together with the clauses

$$\begin{aligned} p_f(X_1, \ldots, X_n, 0, Z) &\leftarrow p_h(X_1, \ldots, X_n, Z), !. \\ p_f(X_1, \ldots, X_n, s(Y), Z) &\leftarrow p_f(X_1, \ldots, X_n, Y, U), p_g(X_1, \ldots, X_n, Y, U, Z), !. \end{aligned}$$

Obviously, P_f is an sld-program with cuts. The desired properties are proven along the same lines as for composition.

Minimalization

Suppose that f is defined by $f(x_1, \ldots, x_n) = \mu y(g(x_1, \ldots, x_n, y) = 0)$ where g is a partial recursive function. By the induction hypothesis, corresponding to g there is an sld-program P_g with cuts and a predicate symbol p_g satisfying the conclusions of the theorem. Define P_f to be P_g together with the clauses

$$\begin{aligned} p_f(X_1, \ldots, X_n, 0) &\leftarrow p_g(X_1, \ldots, X_n, 0, 0), !. \\ p_f(X_1, \ldots, X_n, s(Z)) &\leftarrow r(X_1, \ldots, X_n, Z), p_g(X_1, \ldots, X_n, s(Z), 0), !. \\ r(X_1, \ldots, X_n, 0) &\leftarrow \neg p_g(X_1, \ldots, X_n, 0, 0). \\ r(X_1, \ldots, X_n, s(Z)) &\leftarrow r(X_1, \ldots, X_n, Z), \neg p_g(X_1, \ldots, X_n, s(Z), 0). \end{aligned}$$

Obviously, P_f is an sld-program with cuts. Again, statements 1 and 2 are proven along the same lines as for composition by taking into account the fact that, if p_g occurs in a subgoal of the computation, it is always ground. Note that $r(x_1,\ldots,x_n,z) \in M_{P_f}$ if and only if $(x_1,\ldots,x_n,k) \in \mathrm{dom}(g)$ and $g(x_1,\ldots,x_n,k) \neq 0$ for every $k < z$, and that the goal $r(x_1,\ldots,x_n,Z)$ subsequently yields all answer substitutions Z/z $(z = 0,1,2,\ldots)$ with $(x_1,\ldots,x_n,k) \in \mathrm{dom}(g)$ and $g(x_1,\ldots,x_n,k) \neq 0$ for all $k < z$, which yields the equivalence of 3b and 3c. To show the equivalence of 3a and 3c, note that $P \models r(x_1,\ldots,x_n,z)$ if and only if $P \not\models p_g(x_1,\ldots,x_n,k,0)$ for all $k < z$. So $P \models p_f(x_1,\ldots,x_n,z)$ if and only if $P \models p_g(x_1,\ldots,x_n,z,0)$ and $P \not\models p_g(x_1,\ldots,x_n,k,0)$ for all $k < z$. Now suppose $f(x_1,\ldots,x_n) = z$. Then by the induction hypothesis, the above yields that $P \models p_f(x_1,\ldots,x_n,z)$. Now suppose $f(x_1,\ldots,x_n) \neq z$. We consider three cases:
(1) $g(x_1,\ldots,x_n,z) \neq 0$. Then $P \not\models p_f(x_1,\ldots,x_n,z)$ immediately.
(2) $g(x_1,\ldots,x_n,k) = 0$ for some $k < z$. Again $P \not\models p_f(x_1,\ldots,x_n,z)$ immediately.
(3) $(x_1,\ldots,x_n,k) \notin \mathrm{dom}(g)$ for some $k < z$. Then $r(x_1,\ldots,x_n,k)$ occurs as a subgoal of the computation and, therefore, so does $p_g(x_1,\ldots,x_n,k,0)$. Note that g cannot be one of the basic functions since they are total. For the same reason, g cannot be defined by using composition and primitive recursion on the basic functions only. Consequently, at some point in the computation, a subgoal $p_{f_0}(x_1,\ldots,x_n,y)$ or $p_{f_0}(x_1,\ldots,x_n,Y)$ occurs with $f_0(x_1,\ldots,x_n) = \mu v(g_0(x_1,\ldots,x_n,y) = 0)$ and $(x_1,\ldots,x_n) \notin \mathrm{dom}(f_0)$. There are two subcases to consider:
(i) $g(x_1,\ldots,x_n,m) \neq 0$ for all $m \in \mathbb{N}$. It is easily seen that in this case P_{f_0} will not terminate on the subgoal $p_{f_0}(x_1,\ldots,x_n,Y)$ and will fail on the subgoal $p_{f_0}(x_1,\ldots,x_n,y)$.
(ii) $(x_1,\ldots,x_n,m) \notin \mathrm{dom}(g_0)$ for some $m \in \mathbb{N}$. The condition of this case is exactly as in case (3).
Thus, the argument can be repeated. Since every partial recursive function is defined by using minimalization only finitely often, the conclusion follows by induction. ■

Theorem 2.14 shows that sld-programs with cuts are computationally adequate with respect to SLDNF-resolution as interpreter. It is ongoing work of the authors to investigate the possibilities of developing an alternative interpreter for sld-programs without cuts, based on SLDNF-resolution, which applies inductive methods where necessary.

3 Logic Programs and Dynamical Systems

In this part of the paper, we are going to explore in more detail certain remarks we made in [11] which relate logic programming semantics and dynamical systems. In fact we start by considering, in the context of logic programming, the use of the Hausdorff metric associated with a given metric and connections between logic programming semantics and the Vietoris space of B_P.

3.1 Vietoris Space of B_P

Let (M,ρ) be a metric space. If E,F are subsets of M, see [16], we define the *distance* between E and F to be $\rho(E,F) = \inf\{\rho(x,y); x \in E, y \in F\} = \rho(F,E)$; in case E has only one point x, say, we write $\rho(x,F)$ for $\rho(\{x\},F)$, with a similar convention if F has only one point. Next, let $\mathcal{F}(M)$ denote the set of all non-empty closed subsets of M. For $A,B \in \mathcal{F}(M)$, define $d_A(B) = \max\{\rho(A,y); y \in B\} = \max\{\min\{\rho(x,y); x \in A\}; y \in B\}$. Finally, we define the Hausdorff metric d on $\mathcal{F}(M)$ determined by ρ by setting $d(A,B) = \max\{d_A(B), d_B(A)\}$.

For the duration of this subsection, we will suppose that $\mathcal{L}$ contains at least one function symbol, so that B_P is denumerable. This condition is not essential, but without it the following discussion becomes rather trivial from the topological viewpoint. With this assumption, we may choose a listing $B_P = \{A_0, A_1, A_2, \ldots\}$, of B_P. Now embed B_P into the unit interval $[0, 1]$ by defining $i(A_0) = 0$ and $i(A_n) = 2^{-n}$ for $n \geq 1$. Thus, B_P becomes a compact metric space relative to the metric ρ defined by setting $\rho(A_n, A_m) = |\, i(A_n) - i(A_m)\,|$. Let V_P denote the subspace of I_P consisting of all those elements of I_P which contain A_0. Finally, we recall that the Vietoris space of a topological space X is the set of all non-empty compact subsets of X endowed with the Vietoris (or finite) topology, see [2, 14], and is one of the standard hyperspaces studied in domain theory. In fact, when X is a metric space, the Vietoris topology on X coincides with the topology induced by the Hausdorff metric determined by the metric on X.

3.1 Theorem (1) A subset I of B_P is closed iff it is finite or contains A_0.
(2) The set V_P is a closed, and therefore compact, subset of I_P relative to the topology Q, and each element of V_P is a non-empty closed subset of B_P.
(3) The topology of V_P, as a subspace of I_P endowed with the topology Q, coincides with that induced by the Hausdorff metric determined by the metric ρ on B_P. Therefore, V_P is a closed subspace of the Vietoris space of B_P.

Proof: The details of (1) are elementary and are omitted. For (2), suppose that $I_n \in V_P$ for each $n \in \mathbb{N}$ and that $I_n \to I$ in Q. Since $A_0 \in I_n$ for each n, A_0 is eventually in I_n. Therefore, $A_0 \in I$ by the convergence criterion so that $I \in V_P$ as required. That each element of V_P is a closed subset of B_P follows from (1).

To establish (3), consider the identity map $j : V_P^1 \to V_P^2$, in which V_P^1 denotes V_P endowed with the subspace topology in Q, and V_P^2 denotes V_P endowed with the Hausdorff metric d determined by ρ. Since V_P^1 is compact and V_P^2 is Hausdorff, it suffices to show that j is continuous. To do this, suppose that $I_n \in V_P$ for all $n \in \mathbb{N}$ and that $I_n \to I$ in Q, so that $I \in V_P^1$ by (2) of the theorem. We need to show that $d(I_n, I) \to 0$ as $n \to \infty$ and hence we need to show that both $d_{I_n}(I) \to 0$ and $d_I(I_n) \to 0$ as $n \to \infty$.

Consider $d_{I_n}(I) = \max\{\rho(I_n, y); y \in I\} = \max\{\min\{\rho(x, y); x \in I_n\}; y \in I\}$. Let $\epsilon > 0$ be given, and choose $n_0 \in \mathbb{N}$ so large that $2^{-n_0} \leq \epsilon$. From the fact that $I_n \to I$ in Q, by the convergence criterion and by considering the finitely many elements $A_1, \ldots, A_{n_0}$, there is $k_0 \in \mathbb{N}$ so large that, for $m = 1, \ldots, n_0$, we have:

$$\text{if } A_m \in I, \text{ then for all } n \geq k_0, A_m \in I_n \tag{2}$$

and

$$\text{if } A_m \notin I, \text{ then for all } n \geq k_0, A_m \notin I_n \tag{3}$$

By Equation (2), we now have that, for $n \geq k_0$, $\min\{\rho(x, y); x \in I_n\} = 0$ for any $y \in I \cap \{A_1, \ldots, A_{n_0}\}$. Hence, for $n \geq k_0$, we have $\max\{\min\{\rho(x, y); x \in I_n\}; y \in I\} = \max\{\min\{\rho(x, y); x \in I_n\}; y \in I \text{ and } y = A_{n_0+m} \text{ for some } m \geq 1\}$. But $\min\{\rho(x, y); x \in I_n\} \leq \rho(A_0, y) = i(y)$, since $A_0 \in I_n$ for every n. Hence, for $n \geq k_0$, we have $\max\{\min\{\rho(x, y); x \in I_n\}; y \in I \text{ and } y = A_{n_0+m} \text{ for some } m \geq 1\} \leq \max\{i(y); y \in I \text{ and } y = A_{n_0+m} \text{ for some } m \geq 1\} \leq 2^{-(n_0+1)} < \epsilon$. Therefore, as $n \to \infty$, $d_{I_n}(I) \to 0$, as required.

Now consider $d_I(I_n) = \max\{\rho(x, I); x \in I_n\} = \max\{\min\{\rho(x, y); y \in I\}; x \in I_n\}$. Again, with $\epsilon > 0$ given, and taking the meaning already established for n_0 and k_0, let $n \geq k_0$. Suppose that $x = A_m$, that $x \in I_n$ and that m is any one of the values $1, \ldots, n_0$. Then $y = x$ must also belong to I. Otherwise, if $A_m \notin I$, then by Equation (3) we see that, for all $n \geq k_0$, we have $x \notin I_n$, which is contrary to the choice of $x \in I_n$ with $n \geq k_0$. Hence, again we see that $\min\{\rho(x,y); y \in I\} = 0$ for each $x = A_m \in I_n$ with $m = 1, \ldots, n_0$ and $n \geq k_0$. Thus, for $n \geq k_0$, we have $\max\{\min\{\rho(x,y); y \in I\}; x \in I_n\} = \max\{\min\{\rho(x,y); y \in I\}; x \in I_n$ and $x = A_{n_0+m}$ for some $m \geq 1\}$. But $\min\{\rho(x,y); y \in I\} \leq \rho(x, A_0) = i(x)$, since $A_0 \in I$. Thus, for $n \geq k_0$, we have $\max\{\min\{\rho(x,y); y \in I\}; x \in I_n$ and $x = A_{n_0+m}$ for some $m \geq 1\} \leq \max\{i(x); x \in I_n$ and $x = A_{n_0+m}$ for some $m \geq 1\} \leq 2^{-(n_0+1)} < \epsilon$. Therefore, as $n \to \infty$, we have $d_I(I_n) \to 0$, as required. ∎

3.2 Remark In relation to (2) of Theorem 3.1 it is quite possible in general for a sequence I_n of closed sets to converge in Q to a non-closed set. For example, the sequence in which $I_n = \{A_1, \ldots, A_n\}$, for each n, is a sequence of closed sets converging in Q to $\{A_1, A_2, A_3, \ldots\}$, which is not closed.

Suppose now that P satisfies the condition that it contains at least one unit clause, but is otherwise arbitrary and, in particular, it is not required that T_P be continuous in Q for what follows. This condition just imposed is quite mild and it implies in particular that $T_P(\emptyset) \neq \emptyset$. Moreover, it means that we can arrange matters so that $A_0 \in T_P(I)$ for every $I \in I_P$. Therefore, V_P is an invariant set under T_P in the sense that $T_P(V_P) \subseteq V_P$, and we obtain the following corollary to Theorem 3.1.

3.3 Corollary Suppose that P is any normal logic program which contains at least one unit clause, and that A_0 is chosen so that $A_0 \in T_P(I)$ for every $I \in I_P$. Then, $T_P : V_P \to V_P$. Moreover, since $I_1 = T_P(I)$ belongs to V_P for any $I \in I_P$, iterates of I enter and stay within V_P. Therefore, any fixed point of T_P can be found within V_P. ∎

In the sense that T_P is not usually induced by a point map on B_P, we see that $T_P : V_P \to V_P$ is an abstract dynamical system.

3.4 Example The question clearly arises of providing syntactic conditions under which T_P is a contraction mapping relative to the Hausdorff metric. For example, the "natural numbers" program P as follows:

$$p(o) \leftarrow$$
$$p(s(x)) \leftarrow p(x)$$

has the property that T_P is such a contraction with the obvious listing of B_P. Indeed, T_P has contractivity factor of 2^{-1}, and we sketch the details of this next.

Let $I_1, I_2 \in V_P$ with $I_1 \neq I_2$. Since I_1 and I_2 contain $p(o)$, both $T_P(I_1)$ and $T_P(I_2)$ contain $p(o)$ and $p(s(o))$. Consider $d_{T_P(I_1)}(T_P(I_2)) = \max\{\min\{\rho(x,y); x \in T_P(I_1)\}; y \in T_P(I_2)\}$. If y is $p(o)$ or $p(s(o))$, then $x = y \in T_P(I_1)$ so that $\rho(x,y) = 0$ in these cases and thus $\min\{\rho(x,y); x \in T_P(I_1)\} = 0$. So suppose $y = p(s^k(o)) \in T_P(I_2)$ with $k \geq 2$. Then $\min\{\rho(x,y); x \in T_P(I_1)\}$ is achieved by some $x \in T_P(I_1)$ and is either $\mid 2^{-l} - 2^{-k} \mid$ for some $l \geq 1$ or is $\mid 0 - 2^{-k} \mid$. Suppose the former case holds. Then there is $x = p(s^l(o)) \in T_P(I_1)$ with $l \geq k$. But then $p(s^{l-1}(o)) \in I_1$, $p(s^{k-1}(o)) \in I_2$ and $\min\{\rho(x,y); x \in T_P(I_1)\} = \mid 2^{-l} - 2^{-k} \mid = 2^{-1} \mid 2^{-(l-1)} - 2^{-(k-1)} \mid = 2^{-1} \min\{\rho(x,y'); x \in I_1\}$ with $y' = p(s^{k-1}(o)) \in I_2$. The case when $\min\{\rho(x,y); x \in T_P(I_1)\} = \mid 0 - 2^{-k} \mid$ splits into two subcases each of which gives equalities of the form just derived, and therefore we see that $d_{T_P(I_1)}(T_P(I_2)) = 2^{-1} d_{I_1}(I_2)$. The same sort of argument shows that $d_{T_P(I_2)}(T_P(I_1)) = 2^{-1} d_{I_2}(I_1)$ and hence that $d(T_P(I_1), T_P(I_2)) = 2^{-1} d(I_1, I_2)$ as claimed.

On the other hand, the "even numbers" program:

$$p(o) \leftarrow$$
$$p(s(x)) \leftarrow \neg p(x)$$

does not have this property. For example, taking $I_1 = \{p(o), p(s^2(o)), p(s^5(o))\}$ and $I_2 = B_P$, we find by direct calculation that $d(I_1, I_2) = 2^{-2} = d(T_P(I_1), T_P(I_2))$. Thus, T_P is not in this case a contraction relative to the Hausdorff metric.

3.5 Remark Since I_P is homeomorphic to the Cantor set, $T_P(V_P)$ can be thought of as a subset of the Cantor set and indeed is a compact subset if T_P is continuous in Q. In view of the results of [15], which uses results of Moore to characterize Turing machines as subsets of the Cantor set, one may consider the extent to which $T_P(V_P)$ determines P and the rôle, if any, of continuity of T_P.

3.2 Iterated Function Systems

We begin with a definition.

3.6 Definition An *iterated function system (ifs)*, $\{X; f_1, \ldots, f_n\}$, is given by a finite set of continuous functions $f_i : X \to X$, for $i = 1, \ldots, n$, on a complete metric space (X, d). The ifs is *hyperbolic* if each of the f_i is a contraction mapping.

Now suppose that P is an arbitrary normal logic program and that $P = P_1 \cup \ldots \cup P_n$ is a partition of P into n sub-programs in which the definition of each predicate symbol is contained in one of the P_i (the *definition* of a predicate symbol p is the set of all clauses in P in which the predicate symbol p occurs in the head). We can then write T_P as the union $(\bigcup_{i=1}^n T_{P_i})$ in the sense that for all $I \in I_P$ we have $T_P(I) = (\bigcup_{i=1}^n T_{P_i})(I) = \bigcup_{i=1}^n T_{P_i}(I)$. In this representation, each of the T_{P_i} is to be thought of as a mapping of I_P into itself rather than as a mapping of I_{P_i} into itself.

3.7 Theorem Suppose that P is partitioned as above. Then the following two statements hold.
(1) T_P is continuous in Q at a point $I \in I_P$ iff each of the T_{P_i} is continuous in Q at I.
(2) Suppose that each of the T_{P_i} in the representation above is a contraction relative to Fitting's metric d with contractivity factor $c_i = 2^{-n_i}$, say, where d is determined by the ω-level mapping l. Then T_P is a contraction relative to d with contractivity factor $c = \max\{c_i; i = 1, \ldots, n\}$. Conversely, if T_P is a contraction with factor of contractivity c relative to d, then each of the T_{P_i} is a contraction relative to d with contractivity factor $\leq c$.
Note. In fact, the necessity of (1) and the first conclusion of (2) both hold relative to an arbitrary partition of P, as the proofs given below demonstrate.

Proof: (1) Suppose that each of the T_{P_i} is continuous in Q at I, and that $I_m \to I$ is an arbitrary sequence converging in Q to I. Since $T_{P_i}(I_m) \to T_{P_i}(I)$ for each i, it is an easy consequence of the convergence criterion that $\bigcup_{i=1}^n T_{P_i}(I_m) \to \bigcup_{i=1}^n T_{P_i}(I)$ i.e. that $T_P(I_m) \to T_P(I)$, and so T_P is continuous in Q at I. Conversely, suppose that T_P is continuous in Q at I, that $I_m \to I$ in Q and that $k \in \{1, \ldots, n\}$ is fixed but arbitrary; we show that $T_{P_k}(I_m) \to T_{P_k}(I)$ i.e. that T_{P_k} is continuous in Q at I.

Suppose that $A \in T_{P_k}(I)$. Then $A \in \bigcup_{i=1}^n T_{P_i}(I)$ and hence $A \in T_P(I)$. Since $T_P(I_m) \to T_P(I)$, we see by the convergence criterion that eventually A belongs to $T_P(I_m)$ and hence eventually belongs to $\bigcup_{i=1}^n T_{P_i}(I_m)$. By the nature of the partition $A \notin T_{P_i}(J)$ for any J and any $i \neq k$. Therefore, A is

eventually in $T_{P_k}(I_m)$. Now suppose that $A \notin T_{P_k}(I)$. This case divides into two subcases, the first of which is that $A \notin \bigcup_{i=1}^{n} T_{P_i}(I) = T_P(I)$. Then immediately we have that A eventually not in $T_P(I_m)$ and so A eventually not in $T_{P_k}(I_m)$. For the second subcase, suppose that $A \in T_P(I)$. Thus, $A \in T_{P_j}(I)$ with $j \neq k$. But then immediately $A \notin T_{P_k}(J)$ for any J by the nature of the partition. So A eventually not in $T_{P_k}(I_m)$. Therefore, T_{P_k} is continuous in Q at I by the convergence criterion, as required.
(2) Put $c = \max\{c_i; i = 1, \ldots, n\} = 2^{-m}$, say, where $m \geq 1$. Let $I_1, I_2 \in I_P$ be arbitrary with $I_1 \neq I_2$, and suppose $d(I_1, I_2) = 2^{-k}$. By hypothesis, we have $d(T_{P_i}(I_1), T_{P_i}(I_2)) \leq c_i d(I_1, I_2)$, for $i = 1, \ldots, n$, and so certainly we have, for each i, that $d(T_{P_i}(I_1), T_{P_i}(I_2)) \leq cd(I_1, I_2) = 2^{-m} \cdot 2^{-k} = 2^{-(m+k)}$. Thus, for $i = 1, \ldots, n$, $T_{P_i}(I_1)$ and $T_{P_i}(I_2)$ agree on all ground atoms of level $< m + k$. Consider $T_P(I_1) = \bigcup_{i=1}^{n} T_{P_i}(I_1)$ and $T_P(I_2) = \bigcup_{i=1}^{n} T_{P_i}(I_2)$. Suppose that A is an arbitrary ground atom with $l(A) < m + k$. If $A \in T_P(I_1)$, then $A \in T_{P_j}(I_1)$, say. Hence, $A \in T_{P_j}(I_2)$ and therefore $A \in T_P(I_2)$. Conversely, if $A \in T_P(I_2)$, then $A \in T_P(I_1)$ and so $T_P(I_1)$ and $T_P(I_2)$ agree on all ground atoms A with $l(A) < m + k$. Therefore

$$d(T_P(I_1), T_P(I_2)) \leq 2^{-(m+k)} = cd(I_1, I_2) \tag{4}$$

and so T_P is a contraction with contractivity factor $\leq \max\{c_i; i = 1, \ldots, n\}$. Since $c = c_j$ for some j, there is an atom A with $l(A) = m + k$ and a pair $I_1, I_2 \in I_P$ such that $T_{P_j}(I_1)$ and $T_{P_j}(I_2)$ differ at A. But then $T_P(I_1)$ and $T_P(I_2)$ differ at A, by the conditions on the partition of P, so that c cannot be reduced in Equation (4). In other words, the contractivity factor equals c, as required.

Conversely, suppose that T_P is a contraction with contractivity factor $c = 2^{-m}$. Suppose that $I_1, I_2 \in I_P$ are arbitrary and that $d(I_1, I_2) = 2^{-k}$. Then $d(T_P(I_1), T_P(I_2)) \leq cd(I_1, I_2) = 2^{-(m+k)}$. Thus, $T_P(I_1)$ and $T_P(I_2)$ agree on all ground atoms of level $< m + k$. Fix $i = j \in \{1, \ldots, n\}$. Suppose that A is an arbitrary ground atom with $l(A) < m + k$. If $A \in T_{P_j}(I_1)$, then $A \in T_P(I_1)$ and hence $A \in T_P(I_2)$. Again, by the conditions on the partition of P, this means that $A \in T_{P_j}(I_2)$. Since the converse also holds, we now see that $T_{P_j}(I_1)$ and $T_{P_j}(I_2)$ agree on all ground atoms of level $< m + k$. Therefore, $d(T_{P_j}(I_1), T_{P_j}(I_2)) \leq 2^{-(m+k)} = cd(I_1, I_2)$. It follows that T_{P_j} is a contraction with contractivity factor $\leq c$, as required. ■

Thus, whenever T_P is continuous in Q, $\{I_P; T_{P_1}, \ldots, T_{P_n}\}$ is an iterated function system which is in fact hyperbolic under the conditions of Theorem 3.7 (2).

3.8 Example In the program P:

$$\begin{aligned} q(o) &\leftarrow \\ q(s^3(x)) &\leftarrow p(x) \\ p(o) &\leftarrow \\ p(s^2(x)) &\leftarrow \neg p(x) \end{aligned}$$

the definition of q has contractivity factor $\frac{1}{8}$, and the definition of p has contractivity factor $\frac{1}{4}$. Therefore, P determines a hyperbolic iterated function system with contractivity factor $\frac{1}{4}$.

Supposing, finally, that T_P is continuous in Q, let $F(I_P)$ denote the set of non-empty compact subsets of I_P endowed with the Hausdorff metric d_h induced by d, where d is the metric determined by a finite ω-level mapping l. Then, in the standard way, T_P induces a map $F_P : F(I_P) \to F(I_P)$ defined by $F_P(A) = \{T_P(I); I \in A\}$ which is a contraction with contractivity factor c if T_P is such on I_P. Thus, $F(I_P)$ is the *space of fractals over* I_P and F_P is induced from the iterated function system $\{I_P; T_{P_1}, \ldots, T_{P_n}\}$.

3.3 Generalized Iterated Function Systems

In this final section, we show how the results of the previous section can be extended to arbitrary level mappings l by using the generalized ultrametric d_l in place of Fitting's ultrametric d. We let B denote a countable set, let $D = 2^B$ as defined in Section 2.1 and let $l : B \to \gamma$ be a level mapping where $\gamma > \omega$ is a successor ordinal.

3.9 Definition The *orbit* of $I \in D$ under a function $f : D \to D$ is defined to be the transfinite sequence $(f^\alpha(I))_{\alpha<\gamma}$, where $f^0(I) = I$, $f^\alpha(I) = f(f^{\alpha-1}(I))$ for every successor ordinal $\alpha < \gamma$ and $f^\alpha(I) = \mathrm{gl}((f^\beta(I))_{\beta<\alpha})$ for every limit ordinal $\alpha < \gamma$.

3.10 Proposition With the notation just established, the following statements hold.
(1) $(f^\alpha(I))$ converges in Q iff there exists some ordinal γ_0 such that $f^\alpha(I) = f^\beta(I)$ for all $\alpha, \beta \geq \gamma_0$.
(2) In the situation of (1), $f^{\gamma_0}(I)$ is a fixed point of f.

Proof: Suppose $(f^\alpha(I))$ converges in Q. Then its limit is the set $J = \{a \in B; a \text{ eventually belongs to } f^\alpha(I))\}$. Let $a \in B$ be arbitrary. If $a \in J$, then there is an ordinal $\gamma_a < \gamma$ such that, for all $\alpha \geq \gamma_a$, we have $a \in f^\alpha(I)$. If $a \notin J$, then there is an ordinal $\delta_a < \gamma$ such that, for all $\alpha \geq \delta_a$, we have $a \notin f^\alpha(I)$. Set $\gamma_0 = \sup\{\gamma_a, \delta_a; a \in B\} < \gamma$. Then it is easy to see that $J = f^{\gamma_0}(I) = f^\alpha(I)$ for all $\alpha \geq \gamma_0$, from which the necessity of (1) follows and also (2).

Conversely, suppose γ_0 is an ordinal with the property stated in (1), and let J denote the common value of the $f^\beta(I)$ for all $\beta \geq \gamma_0$. Then it is immediate that $(f^\alpha(I)) \to J$ in Q. ■

3.11 Definition In the situation of Proposition 3.10 (1), we call (D, l, f) a *fixed-point triple* relative to I.

3.12 Lemma Let $f : D \to D$ be strictly contracting with respect to d_l. Then the following statements hold.
(1) For every ordinal α, we have $d_l(f^\alpha(I), f^{\alpha+1}(I)) \leq 2^{-\alpha}$.
(2) Let $a \in B$ with $l(a) = \alpha$. Then either (i) for all $\beta > \alpha$, we have $a \in f^\beta(I)$, or (ii) for all $\beta > \alpha$, we have $a \notin f^\beta(I)$.

Proof: (1) We establish the assertion via transfinite induction. We may suppose that $f^\alpha(I)$ is not a fixed point of f, for that case is trivial. Also, for $\alpha = 0$ the proposition trivially holds. So suppose the proposition holds for all ordinals less than α. If $\alpha = \beta + 1$ is a successor ordinal, we have $d_l(f^\alpha(I), f^{\alpha+1}(I)) < d_l(f^\beta(I), f^{\beta+1}(I)) \leq 2^{-\beta}$. Therefore, $d_l(f^\alpha(I), f^{\alpha+1}(I)) < 2^{-(\alpha-1)}$ and hence $d_l(f^\alpha(I), f^{\alpha+1}(I)) \leq 2^{-\alpha}$ as required. If α is a limit ordinal, we have to show that $d_l(\mathrm{gl}((f^\beta(I))_{\beta<\alpha}), f(\mathrm{gl}((f^\beta(I))_{\beta<\alpha}))) \leq 2^{-\alpha}$. By construction of the greatest limit and the induction hypothesis, we have $d_l(\mathrm{gl}((f^\beta(I))_{\beta<\alpha}), f^{\beta_0}(I)) \leq 2^{-\beta_0}$ for all $\beta_0 < \alpha$. Hence, for every successor ordinal $\beta_0 = \beta_1 + 1 < \alpha$, we have $d_l(f(\mathrm{gl}((f^\beta(I))_{\beta<\alpha})), f^{\beta_0}(I)) < d_l(\mathrm{gl}((f^\beta(I))_{\beta<\alpha}), f^{\beta_1}(I)) \leq 2^{-\beta_1} < 2^{-\beta_0}$. Since d_l is an ultrametric, we get $d_l(\mathrm{gl}((f^\beta(I))_{\beta<\alpha}), f(\mathrm{gl}((f^\beta(I))_{\beta<\alpha}))) \leq 2^{-\beta_0}$ for all $\beta_0 < \alpha$, which suffices.
(2) The proof is by transfinite induction on β. We consider the case $a \in f^{\alpha+1}(I)$, and the case $a \notin f^{\alpha+1}(I)$ is analogous. We have to show that $a \in f^\beta(I)$ for all $\beta > \alpha$. Suppose this is true for all β with $\alpha < \beta < \beta_0$. If $\beta_0 = \beta_1 + 1$ is a successor ordinal, then $d_l(f^{\beta_0}(I), f^{\beta_1}(I)) \leq 2^{-\beta_1}$ and therefore $a \in f^{\beta_0}(I)$ in this case. If β_0 is a limit ordinal, then $a \in f^{\beta_0}(I)$ by definition of greatest limit. ■

3.13 Theorem Let the function $f : D \to D$ be strictly contracting with respect to d_l. Then (D, l, f) is a fixed-point triple relative to every $I \in D$.

Proof: By the previous lemma, $f^\alpha(I)$ converges in Q for every $I \in D$. ∎

The following corollary is immediate.

3.14 Corollary For every strictly level-decreasing program P, (I_P, l, T_P) is a fixed-point triple relative to every $I \in I_P$.

For an injective level mapping, D can be identified with the space of all transfinite sequences $(a_\alpha)_{\alpha<\gamma}$, where $a_\alpha \in \mathbf{2}$ for every α and $\mathbf{2}$ denotes the set $\{0, 1\}$.

3.15 Proposition For every strictly level-decreasing program P there exists an injective level mapping.

Proof: The construction from Defintion 2.1 easily adapts and we omit the details. ∎

The following result gives a straightforward generalization of Theorem 3.7 (2).

3.16 Theorem Let P be an arbitrary normal logic program, let $l : B_P \to \gamma$ be a level mapping and let $P = \bigcup_{\kappa \in K} P_\kappa$ be a partition of P into subprograms. Then the following statements hold.
(1) If every T_{P_κ} is strictly contracting with respect to d_l, then T_P is strictly contracting with respect to d_l.
(2) If the partition of P has the property that the definition of each predicate symbol is contained in one of the P_κ, then the converse of (1) holds.
(3) In the situation of (1) or (2), (I_P, l, T_P) is a fixed-point triple relative to every $I \in I_P$.

Proof: (1) Let $I_1, I_2 \in I_P$ with $d_l(I_1, I_2) = 2^{-\alpha}$, say. We have to show that $T_P(I_1)$ and $T_P(I_2)$ agree on all ground atoms with level $\leq \alpha$. Let $A \in B_P$ with $l(A) \leq \alpha$. If $A \in T_P(I_1)$, then $A \in T_{P_\kappa}(I_1)$, say. By the hypothesis on T_{P_κ} we have $A \in T_{P_\kappa}(I_1)$ and therefore $A \in T_P(I_1)$. Conversely, if $A \in T_P(I_2)$, then $A \in T_P(I_1)$ and so $T_P(I_1)$ and $T_P(I_2)$ agree on all ground atoms A with $l(A) \leq \alpha$.
(2) Let $I_1, I_2 \in I_P$ with $d_l(I_1, I_2) = 2^{-\alpha}$. Then $T_P(I_1)$ and $T_P(I_2)$ agree on all ground atoms of level $\leq \alpha$. Let $A \in B_P$ with $l(A) \leq \alpha$. If $A \in T_{P_\kappa}(I_1)$, then $A \in T_P(I_1)$ and hence $A \in T_P(I_2)$. By the hypothesis concerning the partition we get $A \in T_{P_\kappa}(I_2)$. Since the converse also holds, we now see that $T_{P_\kappa}(I_1)$ and $T_{P_\kappa}(I_2)$ agree on all ground atoms of level $\leq \alpha$, which suffices.
(3) This follows immediately from Theorem 3.13. ∎

Conclusions

We have shown that the class of strictly level-decreasing logic programs is of special interest amongst all logic programs in that it has several rather pleasing properties, as follows. First, each program in this class has a unique supported model and therefore there is no argument about which model is "best" since all the standard models (perfect model, weakly perfect model, stable model etc.) coincide. Second, this class is computationally adequate in that there is a rather simple interpreter relative to which it can compute all the partial recursive functions. Finally, we have shown that there are interesting aspects of this class of programs which relate it to ideas of current interest in dynamical systems. Such ideas are actively being pursued in the context of domain theory and in real number computation by Edalat in [2] and in more recent work of his. The full extent of the connections between logic programming and these other areas just mentioned remains to be determined, and is actively under investigation by the present authors.

References

[1] Apt KR, Pedreschi D. Reasoning about termination of pure prolog programs. Information and Computation 1993; 106:109-157

[2] Edalat A. Dynamical systems, measures and fractals via domain theory. Information and Computation 1995; 120 (1):32-48

[3] Fitting M. Metric methods: three examples and a theorem. J. Logic Programming 1994; 21 (3):113-127

[4] Hitzler P. Topology and logic programming semantics. Diplomarbeit in Mathematik, Universität Tübingen, 1998

[5] Lloyd JW. Foundations of logic programming, Second Edition. Springer-Verlag, Berlin, 1988

[6] Priess-Crampe S, Ribenboim P. Fixed points, combs and generalized power series. Abh. Math. Sem. Univ. Hamburg 1993; 63:227-244

[7] Priess-Crampe S, Ribenboim P. Ultrametric spaces and logic programming. Preprint, October 1997; pp 1-12

[8] Przymusinski T. On the declarative semantics of deductive databases and logic programs. In: Minker J (ed) Foundations of deductive databases and logic programming. Morgan Kaufmann Publishers Inc., Los Altos, 1988, pp 193-216

[9] Seda AK. Topology and the semantics of logic programs. Fundamenta Informaticae 1995; 24 (4):359-386

[10] Seda AK. Quasi-metrics and the semantics of logic programs. Fundamenta Informaticae 1997; 29 (1):97-117

[11] Seda AK, Hitzler P. Topology and iterates in computational logic. In: Proceedings of the 12th Summer Conference on Topology and its Applications: Special Session on Topology in Computer Science, Ontario, August 1997; pp 1-43 (Topology Proceedings Vol 22. To appear)

[12] Seda AK, Hitzler P. Sur les programmes logiques lacalement stratifiés. C.R. Acad. Sc. Paris 1998; pp 1-5. To appear

[13] Šebelík J, Štěpánek P. Horn clause programs for recursive functions. In: Clark KL, Tärnlund S-Å (eds) Logic programming. Academic Press, New York, 1982, pp 324-340

[14] Smyth MB. Topology. In: Abramsky S, Gabbay DM, Maibaum TSE (eds) Handbook of logic in computer science, Vol. 1. Oxford Science Publications, Oxford University Press, 1992, pp 641-761

[15] Walsh J. Topology, fractals and domain theory – towards a geometria. M.Sc Thesis, University of Dublin, Trinity College, 1997

[16] Willard S. General topology. Addison Wesley Series in Mathematics, Addison Wesley Publishing Company Inc., Massachusetts, 1970