Generalized Metrics and Uniquely Determined Logic Programs*

Pascal Hitzler†

Artificial Intelligence Institute
Department of Computer Science
Dresden University of Technology
Dresden, Germany
phitzler@inf.tu-dresden.de
www.wv.inf.tu-dresden.de/~pascal/

Anthony Karel Seda

Department of Mathematics
National University of Ireland
University College Cork
Cork, Ireland
a.seda@ucc.ie
maths.ucc.ie/staff/seda/

November 28, 2001

Abstract

The introduction of negation into logic programming brings the benefit of enhanced syntax and expressibility, but creates some semantical problems. Specifically, certain operators which are monotonic in the absence of negation become non-monotonic when it is introduced, with the result that standard approaches to denotational semantics then become inapplicable. In this paper, we show how generalized metric spaces can be used to obtain fixed-point semantics for several classes of programs relative to the supported model semantics, and investigate relationships between the underlying spaces we employ. Our methods allow the analysis of classes of programs which include the acyclic, locally hierarchical, and acceptable programs, amongst others, and draw on fixed-point theorems which apply to generalized ultrametric spaces and to partial metric spaces.

*To appear in: Proceedings of the Dagstuhl Seminar 00231, Topology in Computer Science: Constructivity; Asymmetry and Partiality; Digitalization, Schloss Dagstuhl, Germany, June, 2000. Special issue of *Theoretical Computer Science*, Elsevier, 2001.

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

Contents

1 Introduction **2**

2 Preliminaries **5**
2.1 The Supported Model Semantics 5
2.2 Uniquely Determined Programs 6
2.3 Generalized Metrics 11
2.4 The Query and Atomic Topologies 13

3 Basic Construction **13**

4 Dislocated Metrics and the Φ_ω^*-Accessible Programs **17**
4.1 The Fixed-Point Theorem of Matthews 17
4.2 An Application to the Φ_ω^*-Accessible Programs 18
4.3 Dislocated Metrics and Metrics 22

5 Dislocated Generalized Ultrametrics and the Φ^*-Accessible Programs **24**
5.1 A Generalized Priess-Crampe & Ribenboim Fixed-Point Theorem 24
5.2 Application to the Φ^*-Accessible Programs 26
5.3 Generalized Ultrametrics and Dislocated Generalized Ultrametrics 28

6 Φ-Accessible Programs **30**

7 Related and Further Work **31**

8 Conclusions **33**

A Appendix: More on the Atomic Topology **38**

1 Introduction

In recent years, the role of topology in Logic Programming has come to be recognized, with applications of methods of topology to several areas within logic programming including: continuous models of computation, building formal models of hybrid systems, modularity of programs, fixed-point theory, inductive logic programming, studies in termination and verification, connections between logic programming and neural networks, applications to disjunctive databases, and construction of standard models of logic programs. Whilst the bibliography of this paper is not in any way intended to be comprehensive, the reader may care to consult [4, 5, 6, 7, 16, 19, 20, 22, 23, 29, 31, 35, 41, 43] for some sample results and to gain an overview; a brief discussion of some of these works is to be found in Section 7 of this paper. In particular, it is now appreciated that topological methods can be employed to obtain fixed-point semantics for logic programs in situations where methods based on order may fail. In this paper, we will pursue the observation just made quite extensively by examining the use of fixed-point theorems which utilize various types of generalized metric. In addition, we show how these latter theorems may be applied with the specific aim of establishing that certain classes of programs are uniquely determined in the sense that each member of the class has a unique supported model, that is, an unambiguous Clark completion semantics.

In the classical approach to logic programming semantics, see [32], one associates with each definite or positive program[1] an operator T_P called the *single-step* or *immediate consequence operator*, see Section 2. This operator turns out to be Scott-continuous on the complete lattice of all interpretations. An application of the well-known fixed-point theorem for continuous operators on complete partial orders yields a least fixed point of T_P. Usually, one takes this least fixed point to be the denotational semantics, or meaning, of the program in question, and indeed, in the definite case, it turns out that this semantics agrees very well with the procedural and logical reading of the program, see [32].

However, when syntax is enhanced by introducing classical negation to obtain the so-called *normal* logic programs, the single-step operator becomes non-monotonic, and hence not Scott-continuous, in general. This fact has the unfortunate consequence that the classical approach described above using the fixed-point theorem for Scott-continuous mappings becomes invalid, and other methods have to be sought. To date, these include: (1) restricting the syntax of the programs in question (see for example [1, 10, 37, 43]), (2) using alternative operators (see for example [15, 17, 18, 21]), and (3) applying alternative fixed-point theorems which apply to non-monotonic operators. It is this latter point (3) which we address here.

The main alternative to the Knaster-Tarski theorem and its relatives, such as the fixed-point theorem already alluded to above, is the Banach contraction mapping theorem for complete metric spaces. In some cases, for example for the acyclic[2] programs, the Banach theorem can indeed be applied, see [16, 29, 44]. Acyclic programs, however, are a rather restrictive class and, furthermore, the topological spaces which arise in the area of denotational semantics are often not metrizable. It is therefore of interest to find fixed-point theorems for spaces which are weaker than metrics in a topological sense. The options include (a) quasi-metrics, see [42, 46], which have a well-established presence within domain theory,

[1] A program in which negation does not occur.

[2] Called *ω-locally hierarchical* in [10].

(b) generalized ultrametric spaces having arbitrary partially ordered sets as the codomain of their distance function, see [31, 43, 35, 22], and (c) partial metrics, or the slightly more general dislocated metrics[3], which differ from metrics in that one allows the distance from a point to itself to be non-zero, see [24].

Here, we will see that the theorem of Priess-Crampe and Ribenboim on generalized ultrametrics, the fixed-point theorem of Matthews on dislocated metrics, and a theorem which is obtained by merging those two can be employed in order to analyse logic programs from the point of view of denotational semantics. Preliminary results along these lines have already been obtained by the authors in [22, 23, 24, 25, 26] and presented at various conferences and workshops. In this paper, these results are placed in a general framework and further results and new proofs are included. Where proofs have already been published, references only are given so that the paper is partly a survey of our results in this area.

The plan of the paper is as follows. After some preliminaries in Section 2, the basic underlying construction of (generalized) metric is presented in Section 3 and applied to both acyclic and locally hierarchical programs using the Banach contraction mapping theorem respectively the theorem of Priess-Crampe and Ribenboim. This line of thinking is then extended in Sections 4 and 5 in order to permit the investigation of larger classes of programs, first by employing dislocated metrics and the Matthews fixed-point theorem, and then by merging dislocated metrics and generalized ultrametrics. An even more general approach is then considered in Section 6, where we will ultimately employ our results in order to study the class of all programs which have a total Kripke-Kleene semantics [15]. At all times, investigations of the underlying spaces and the interrelationships between them will go hand in hand with the results on logic programming semantics which are the motivation for these investigations. Finally, in Appendix A, we collect together some background facts concerning the topology which underlies all the work presented here.

Acknowledgements. We thank Michel Schellekens for bringing [34] to our attention, Keye Martin for pointing us to [12], and Pawel Waszkiewicz for providing us with some facts about partial metric spaces. Roland Heinze made us aware of an incorrecteness in an earlier version of Definition 2.2. The suggestions of three anonymous referees have helped us to substantially improve the presentation of the paper in several places. Finally, we wish to thank the organizers of the Dagstuhl Seminar 00231 for inviting us to present our work at that meeting.

[3] Called *metric domains* in [33].

2 Preliminaries

2.1 The Supported Model Semantics

A (*normal*) *logic program* is a finite set of universally quantified clauses of the form

$$\forall\,(A \leftarrow L_1 \wedge \cdots \wedge L_n)\,,$$

where A is an atom and all the L_i are literals, usually written simply as

$$A \leftarrow L_1, \ldots, L_n.$$

We call A the *head* of the clause and $L_1, \ldots, L_n$ (which denotes $L_1 \wedge \ldots \wedge L_n$) the *body* of the clause. Each L_i is called a *body literal* of the clause. A program is called *definite* if no negation symbol occurs in it. If p is a predicate symbol occurring in P, then the *definition* of p consists of all clauses from P whose head contains p.

For a given logic program P, we denote the Herbrand base (the set of all ground atoms in the underlying first order language) by B_P. As usual, (Herbrand-) interpretations of P will be identified with subsets of B_P, so that the power set $I_P = 2^{B_P}$ is the set of all interpretations of P. The set of all ground instances of each clause in a program P will be denoted by ground(P). A *level mapping* is a function $l : B_P \to \alpha$, where α is an arbitrary (countable) ordinal; we call the value $l(A)$ the *level* of the element A of B_P. We always assume that a level mapping is extended to ground literals by setting $l(\neg A) = l(A)$ for each $A \in B_P$. If $\alpha = \omega$, the smallest infinite ordinal, we call l an *ω-level mapping*. We identify ω with the set of natural numbers, $\mathbb{N}$.

A standard approach to logic programming semantics, that is, to assigning a reasonable meaning to a given logic program, is to identify models of the program which have certain additional properties. We will focus here on the *supported model semantics* or *Clark completion semantics*. To do this, we define the *immediate consequence operator*, or *single-step operator*, T_P, for a given logic program P as a mapping $T_P : I_P \to I_P$ of interpretations to interpretations as follows: $T_P(I)$ is the set of all $A \in B_P$ for which there exists an element $A \leftarrow L_1, \ldots, L_n$ of ground(P), with head A, satisfying $I \models L_1 \wedge \cdots \wedge L_n$. Note that T_P is in general not monotonic. As it turns out, the models of P are exactly the pre-fixed points of T_P, and hence are those interpretations I which satisfy $T_P(I) \subseteq I$. A *supported model* (or model of the Clark completion) of P is a fixed point of T_P.

The Clark completion of a program was introduced in [11], see also [32], as a way of interpreting logic programming clauses as equivalences rather than as implications. Clark studied the relationship between his completion and the interpretation of negation as finite failure, which is the way negation is treated for example in Prolog. First, a given program P is *completed* to obtain a set $\text{comp}(P)$ of logical formulas. The models of $\text{comp}(P)$ are then taken to be the declarative semantics of P. It not only turns out [11] that negation as failure is correct with respect to this semantics, but also that the models of $\text{comp}(P)$ can be obtained, using a simple identification, as the fixed points of the operator T_P introduced above. These fixed points, which are models of P, were termed *supported* in [1] since such a model I provides support for belief in each ground atom A with $I \models A$ in the following precise sense: if $I \models A$, then there is a ground instance $A \leftarrow L_1, \ldots, L_n$ of a clause in P such that $I \models L_1 \wedge \cdots \wedge L_n$; thus, each ground atom A which is true in I is true for a reason provided by the program and I itself.

In the sequel, we will study the declarative reading of logic programs as given by the Clark completion semantics, or supported model semantics. This undertaking necessarily inherits any limitations which are implicit in the Clark completion semantics itself, and therefore can only be as satisfactory as this semantics.[4] However, the declarative reading we have chosen to work with provides the conceptual clarity which is needed to understand the central aspects of our approach, namely to use generalized metrics for the study of non-monotonic semantic operators in logic programming. Certainly, this paper is not an end in itself.

2.2 Uniquely Determined Programs

We next introduce some classes of logic programs which will be studied in detail in the sequel. We will see later that all these programs are *uniquely determined* in that each of them has a unique supported model. The following definition is taken from [2] where it was employed in defining acceptable programs, much studied in the context of termination analysis. It is the inspiration for the more general classes which will be given in Definition 2.2.

2.1 Definition Let P be a logic program and let p and q be predicate symbols occurring in P.

1. p *refers to* q if there is a clause in P with p in its head and q in its body.
2. p *depends on* q if (p, q) is in the reflexive, transitive closure of the relation *refers to*.
3. Neg_P denotes the set of predicate symbols in P which occur in a negative literal in the body of a clause in P.

[4] Indeed, although there has been considerable progress concerning the study of declarative, model-theoretic semantics of logic programs, this quest is still inconclusive for programs with negation.

4. Neg_P^* denotes the set of all predicate symbols in P on which the predicate symbols in Neg_P depend.

5. P^- denotes the set of clauses in P whose head contains a predicate symbol from Neg_P^*.

2.2 Definition A program P is called Φ^*-*accessible* if and only if there exists a level mapping l for P and a model I for P which is a supported model of P^- such that the following condition holds. For each clause $A \leftarrow L_1, \ldots, L_n$ in ground(P), we either have $I \models L_1 \wedge \cdots \wedge L_n$ and $l(A) > l(L_i)$ for all $i = 1, \ldots n$ or there exists $i \in \{1, \ldots, n\}$ such that $I \not\models L_i$ and $l(A) > l(L_i)$. Furthermore, P is called Φ_ω^*-*accessible* if it is Φ^*-accessible and l can be taken to be an ω-level mapping.

Next, P is called Φ-*accessible* if and only if there exists a level mapping l for P and a 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 $I \models L_1 \wedge \cdots \wedge L_n$ and $l(A) > l(L_i)$ for all $i = 1, \ldots, n$.

(ii) $I \not\models A$ and 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 $I \not\models L_i$ and $l(A) > l(L_i)$.

Again, P is called Φ_ω-*accessible* if it is Φ-accessible and l can be taken to be an ω-level mapping.

Finally, a program P is called *locally hierarchical*, see [10], if there exists a level mapping $l : B_P \to \alpha$ such that for each clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) and for all $i = 1, \ldots, n$ we have $l(A) > l(L_i)$. If α can be chosen to be ω, then P is called *acyclic*.

We note that if a program is Φ-accessible with respect to a model I and a level mapping l, then I is supported. This follows easily from the defining conditions.

Relationships between the classes of programs defined above are represented in Figure 1, where an arrow pointing from one class to a second indicates inclusion of the first inside the second. We also note that both the class of Φ^*-accessible and the class of Φ-accessible programs contain the acceptable programs of [2] and the locally hierarchical programs.

Examples

We illustrate the definitions above by means of some example programs which can actually be run under Prolog. The language which we will consider contains a constant symbol 0 and a function symbol s. The intended meaning of these symbols is that 0 stands for the natural number zero, and s stands for the successor

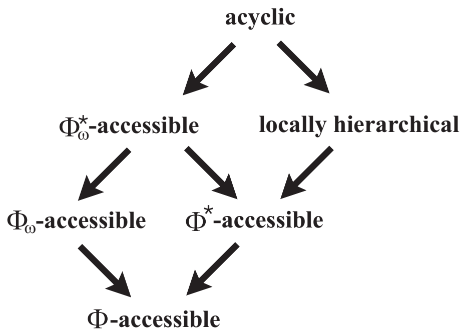

Figure 1: Relationships between unique supported model classes

function on the natural numbers. We abbreviate $s(\ldots(s(0))\ldots)$, in which s occurs n times, by $s^n(0)$, and we think of this as the natural number n. Variables are denoted by uppercase letters as usual under Prolog.

The following program `even` is acyclic:

$$\begin{aligned} \mathit{even}(0) &\leftarrow \\ \mathit{even}(s(X)) &\leftarrow \neg \mathit{even}(X) \end{aligned}$$

This program can be used to check whether a given term $s^n(0)$ represents an even number, but it cannot be used to generate the even numbers because the call `?-`$\mathit{even}(X)$ fails immediately since the derived goal `?-`$\neg\mathit{even}(Y)$ flounders[5]. In order to generate the even numbers, we use the so-called *generate-and-test scheme* as in the following program `getEven`, which consists of the program `even` together with the following clauses:

$$\begin{aligned} \mathit{nn}(0) &\leftarrow \\ \mathit{nn}(s(X)) &\leftarrow \mathit{nn}(X) \\ \mathit{getEven}(X) &\leftarrow \mathit{nn}(X), \mathit{even}(X) \end{aligned}$$

In this program, the predicate *nn* is used in order to generate, successively, all natural numbers. As soon as one of them is generated, it is tested for being even by invoking a call to *even*. The program call `?-`$\mathit{getEven}(X)$ successively generates all even numbers.

Program `getEven` is still acyclic with respect to the level mapping which maps each ground atom to the natural number which equals the number of occurrences of the function symbol s in the atom. The following program `existsEven` is not

[5]It flounders because it attempts to evaluate a negated atom in which a variable occurs. This cannot be resolved under Prolog.

acyclic, but is locally hierarchical, and consists of the definitions of *even*, *nn* and the following clause:

$$existsEven \leftarrow nn(X), even(X)$$

Procedurally, the call `?-`*existsEven* succeeds if and only if there exists an even natural number. Admittedly, this example is somewhat academic, but it illustrates why it may be interesting to study locally hierarchical programs which are not acyclic. For example, the predicate *nn* could be replaced by the generator of some more complicated data structure, and *even* by a sophisticated test predicate. The resulting program is then no longer trivial like `existsEven`, but it is still locally hierarchical if the subprograms belonging to the substituted predicates are.

We note that the clause defining *existsEven* contains variables which occur in the body of the clause but not in its head. Such variables are called *local*. Locally hierarchical programs without local variables turn out to be acyclic, see [44].

As an example of a Φ_ω^*-accessible program we use the following program `game` from [2] as follows. Let G be an acylic finite graph.

$$\begin{aligned} win(X) &\leftarrow move(X,Y), \neg win(Y) \\ move(a,b) &\leftarrow \qquad \text{for all } (a,b) \in G \end{aligned}$$

It was shown in [2] that `game` is acceptable, and the proof carries over directly to show that it is Φ_ω^*-accessible.

Acyclic programs always terminate, see [3], and are therefore not expressive enough for implementing all partial recursive functions. We give next a more sophisticated example concerning the minimalization step in implementing partial recursive functions. The resulting program will turn out to be Φ^*-accessible.

Suppose that g is a partial recursive function (on natural numbers in successor notation) with $n+1$ arguments, and that the partial recursive function f is defined by $f(x_1,\dots,x_n) = \mu y(g(x_1,\dots,x_n,y) = 0)$, that is, $f(x_1,\dots,x_n)$ is the least y such that $g(x_1,\dots,x_n,y) = 0$ and $g(x_1,\dots,x_n,z)$ is not undefined for all $z < y$, if such a y exists, and is undefined otherwise. Suppose, furthermore, that P_g is a Φ^*-accessible program which defines an $(n-2)$-ary predicate p_g such that a call `?-`$p_g(x_1,\dots,x_n,y,U)$ does not terminate if $g(x_1,\dots,x_n,y)$ is undefined, and otherwise yields the computed answer substitution which gives the value $g(x_1,\dots,x_n,y)$ to U. Now consider the following program P_f which contains[6]

[6]We assume without loss of generality that P_g does not contain predicate symbols p_f, r and *lt*.

P_g and the following clauses:

$$\begin{aligned}
p_f(X,Y) &\leftarrow p_g(X,0,U), r(X,0,U,Y)\\
r(X,Y,0,Y) &\leftarrow\\
r(X,Y,s(V),Z) &\leftarrow p_g(X,s(Y),U), r(X,s(Y),U,Z), lt(Y,Z)\\
lt(0,s(X)) &\leftarrow\\
lt(s(X),s(Y)) &\leftarrow lt(X,Y)
\end{aligned}$$

Program P_f is an adaptation of a program used in [32, Theorem 9.6][7] to show that every partial recursive function can be implemented by a definite program. Using the proof in [32], it is easy to see that P_f indeed implements f, and the details of this are not our main concern here. We will instead show that P_f is Φ^*-accessible under some reasonable assumptions on P_g. However, we delay this discussion until the end of Section 5.2, since the analysis of the program P_f is easier once we have gained some insight into the nature of Φ^*-accessible programs. The considerations just given lead to the fact reported in [21] that every partial recursive function can be implemented, under Prolog, by a definite Φ^*-accessible program. We know of no work describing a smaller class of programs with this property.

Uniquely Determined Programs in Context

Some of the classes of programs which satisfy Definition 2.2 are already well-known in the literature, since level mappings have provided a convenient tool for studying termination properties. Bezem in [3] showed that the acyclic programs are exactly the programs which terminate under any selection rule. Apt and Pedreschi in [2] relaxed the notion of acyclicity and obtained the class of *acceptable* programs which, in the absence of floundering, correspond exactly to left-termination, that is, to termination under the left-to-right selection rule as implemented in Prolog. Both classes are contained in the class of all Φ^*_ω-accessible programs. In fact, each Φ^*_ω-accessible program is acceptable modulo reordering of body literals (in ground(P)), and can therefore be understood, in the absence of floundering, as corresponding to a terminating program under a *don't know* non-deterministic selection rule. We do not study acceptable programs here, as such, since they are completely subsumed, for our purposes, by the more general classes we have introduced in Definition 2.2.

The Φ-accessible programs are exactly the programs which have a total *Kripke-Kleene semantics* as introduced by Fitting in [15]. This follows from the results presented in [21] where the present authors proposed a unifying view of certain

[7] The original program of Lloyd is not Φ^*-accessible.

classes of programs, including those defined above, based on various three-valued logics. Building on [21], we subsequently showed in [25, 27] that each Φ-accessible program is *weakly stratified* in the sense of Przymusinska and Przymusinski [36]. This fact strengthens what is in effect the well-known result from [17] that every Φ-accessible program has a total well-founded semantics and therefore a unique stable model [18], although the terminology "Φ-accessible" was not of course used in these earlier works. We note that Definition 2.2 differs slightly from the one used in [21, 25, 27], but equivalence between the respective definitions can easily be established.

For the study of the termination properties and relationships mentioned in this section we refer the reader to the literature already indicated, and we proceed now to the study of generalized metrics in logic programming.

2.3 Generalized Metrics

Metrics, and their generalized versions as introduced below, provide an abstract quantitative measure of *distance* between points in a space. In the following, we collect together the different definitions of generalized metrics which will be used in the sequel.

2.3 Definition Let X be a set and let $d : X \times X \to \mathbb{R}_0^+$ be a function, called a *distance function*, where $\mathbb{R}_0^+$ denotes the set of non-negative real numbers. Consider the following conditions:

(Mi) For all $x \in X$, $d(x,x) = 0$.

(Mii) For all $x, y \in X$, if $d(x,y) = 0$, then $x = y$.

(Miii) For all $x, y \in X$, $d(x,y) = d(y,x)$.

(Miv) For all $x, y, z \in X$, $d(x,y) \leq d(x,z) + d(z,y)$.

(Miv$'$) For all $x, y, z \in X$, $d(x,y) \leq \max\{d(x,z), d(z,y)\}$.

Terminology for d, depending on which of the conditions it satisfies, is given in Table 1, where a "$\times$" indicates that the respective condition holds, and will be extended to the pair (X,d) in the obvious way. Thus, for example, (X,d) will be called a metric space if d is a metric, and so on. In fact, it will often be convenient throughout the paper to abuse notation and refer to properties of d which, more properly, are properties of (X,d), and vice versa.

The most important of the notions just defined is that of metric, and indeed results on metric spaces can be found in any standard textbook on general topology, see for example [48]. One of the central results in the theory is the Banach

Name	(Mi)	(Mii)	(Miii)	(Miv)	(Miv′)
metric	×	×	×	×	
ultrametric	×	×	×	×	×
dislocated metric (*d-metric*)		×	×	×	
dislocated ultrametric (*d-ultrametric*)		×	×	×	×

Table 1: Metric Definitions

contraction mapping theorem which we state for convenient reference. Note that sequences such as $(x_n)_{n\in\mathbb{N}}$ will usually be denoted simply by (x_n) where no confusion is caused.

A sequence (x_n) in a metric space (X, d) *converges with respect to* d (or *in* d) if there exists $x \in X$ such that $d(x_n, x)$ converges in the real line to 0 as $n \to \infty$. In this case, x is called the *limit* of (x_n). A sequence (x_n) in a metric space is called a *Cauchy sequence* if, for each $\varepsilon > 0$, there exists $n_0 \in \mathbb{N}$ such that for all $m, n \geq n_0$ we have $d(x_m, x_n) < \varepsilon$. A metric space X is called *complete* if every Cauchy sequence in X converges. A function $f : X \to X$ is called a *contraction* if there exists $0 \leq \lambda < 1$ such that the inequality $d(f(x), f(y)) \leq \lambda d(x, y)$ holds for all $x, y \in X$.

2.4 Theorem (Banach) Let (X, d) be a complete metric space and let $f : X \to X$ be a contraction. Then f has a unique fixed point x_0 which can be obtained as the limit of the sequence $(f^n(x))_{n\in\mathbb{N}}$ for any choice of $x \in X$.

Dislocated metrics were introduced in [33], and we will return to them later.

An alternative method of obtaining generalized metrics is by relaxing the conditions on the codomain of the distance function: we allow arbitrary partially ordered sets with least element in place of the set of real numbers. Such generalized metric spaces originate from valuation theory and were introduced to logic programming in [35], see also [22].

2.5 Definition Let X be a set and let $(\Gamma, \leq)$ be a partially ordered set with least element 0. We call (X, d, Γ), or simply (X, d), a *generalized ultrametric space* (*gum*) if $d : X \times X \to \Gamma$ is a function satisfying the following conditions for all $x, y, z \in X$ and $\gamma \in \Gamma$:

(Ui) $d(x, y) = 0$ implies $x = y$.

(Uii) $d(x, x) = 0$.

(Uiii) $d(x, y) = d(y, x)$.

(Uiv) If $d(x,y) \leq \gamma$ and $d(y,z) \leq \gamma$, then $d(x,z) \leq \gamma$.

If d satisfies conditions (Ui), (Uiii) and (Uiv) only, then we call (X,d) a *dislocated generalized ultrametric space* (*d-gum*). A *ball* in a (dislocated) generalized ultrametric space (X,d) is a set of the form $B_\gamma(x) = \{y \in X \mid d(x,y) \leq \gamma\}$, where $\gamma \in \Gamma$; we call x a *midpoint* of the ball (note that any point of a ball in a generalized ultrametric space is a midpoint, see [35] and Lemma 5.2), and we call γ its *radius*. Similar terminology applies in metric spaces also. We note at this point that a ball in a d-gum may be empty. A (dislocated) generalized ultrametric space X is called *spherically complete* if the intersection of each chain[8] of (non-empty) balls is non-empty.

2.4 The Query and Atomic Topologies

We will need to make use of the atomic topology Q which was introduced in [41]. This topology is a generalization of the query topology discussed in [7]. In fact, we will only use two properties of this topology, namely, that limits in Q are unique, and that limits of sequences in Q are characterized as follows.

2.6 Proposition A sequence (I_n) converges in Q iff for every $A \in B_P$ we have either $A \in I_n$ eventually or eventually $A \notin I_n$, that is, for every $A \in B_P$ there exists some n_0 such that either for all $n \geq n_0$ we have $A \in I_n$ or for all $n \geq n_0$ we have $A \notin I_n$. Moreover, if (I_n) converges in Q, then its limit is the set $\{A \in B_P \mid A \text{ eventually belongs to } I_n\}$.

We also note that it is possible to characterize Q using logical notions. For the reader with some topological background we have recorded some further results and observations about Q, which are independent of the rest of the paper, in Appendix A.

3 Basic Construction

The following construction is fundamental to our investigations, and was already hinted at in [16]. It is also closely related to a definition used in [46] for obtaining quasi-metrics from domains, see also [43].

Let P be a logic program and let $l : B_P \to \gamma$ be a level mapping for P. Let $\Gamma = \{2^{-\alpha} \mid \alpha \leq \gamma\}$, ordered by $2^{-\alpha} < 2^{-\beta}$ iff $\beta < \alpha$, and denote $2^{-\gamma}$ by 0.

3.1 Definition We define a function $d : I_P \times I_P \to \Gamma$ by setting $d(I,J) = 0$ if $I = J$, and, when $I \neq J$, by setting $d(I,J) = 2^{-\alpha}$, where I and J differ on some ground atom of level α but agree on all ground atoms of lower level.

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

It turns out that (I_P, d) is a spherically complete generalized ultrametric space, see [43]. If $\alpha = \omega$, then (I_P, d) is a complete ultrametric space.

3.2 Definition A function $f : X \to X$ defined on a generalized ultrametric space (X, d) is called *strictly contracting* if $d(f(x), f(y)) < d(x, y)$ for all $x, y \in X$ with $x \neq y$.

3.3 Theorem Let X be a spherically complete generalized ultrametric space and let $f : X \to X$ be strictly contracting. Then f has a unique fixed point.

This theorem, due to Priess-Crampe and Ribenboim [35], is clearly analogous to the Banach contraction mapping theorem for complete metric spaces. The relationship between these two theorems can be clarified using the following proposition, and we note that every (conventional) ultrametric space is also a generalized ultrametric space.

3.4 Proposition Let (X, d) be an ultrametric space. If X is spherically complete, then it is complete. The converse does not hold in general.

Proof: Assume that (X, d) is spherically complete and that (x_n) is a Cauchy sequence in (X, d). Then, for every $k \in \mathbb{N}$, there exists a least $n_k \in \mathbb{N}$ such that for all $n, m \geq n_k$ we have $d(x_n, x_m) \leq \frac{1}{k}$. We note that n_k increases with k. Now consider the set of balls $\mathcal{B} = \{B_{\frac{1}{k}}(x_{n_k}) \mid k \in \mathbb{N}\}$. By (Uiv), $\mathcal{B}$ is a decreasing chain of balls and has non-empty intersection B by the spherical completeness of (X, d). Let $a \in B$. Then it is easy to see that (x_n) converges to a (and hence that $B = \{a\}$ is a one-point set since limits in (X, d) are unique) and therefore (X, d) is complete.

In order to show that the converse does not hold in general, define an ultrametric d on $\mathbb{N}$ as follows. For $n, m \in \mathbb{N}$, let $d(n, m) = 1 + 2^{-\min\{m,n\}}$ if $n \neq m$ and set $d(n, n) = 0$ for all $n \in \mathbb{N}$. The topology induced by d is then the discrete topology on $\mathbb{N}$, and the Cauchy sequences with respect to d are exactly the sequences which are eventually constant. So, $(\mathbb{N}, d)$ is complete. Now consider the chain of balls B_n of the form $\{m \in \mathbb{N} \mid d(m, n) \leq 1 + 2^{-n}\}$. Then we obtain $B_n = \{m \mid m \geq n\}$ for all $n \in \mathbb{N}$ and hence $\bigcap B_n = \emptyset$. ■

We note that the immediate consequence operator T_P is not in general strictly contracting if no conditions are placed on the program P. However, for locally hierarchical programs P, T_P is strictly contracting. Hence, by Theorem 3.3 we can conclude that each locally hierarchical program has a unique supported model (as shown first in [10] by completely different methods). For details of the proofs, we refer to [22]. In fact, if P is acyclic, then the contraction mapping theorem itself can be applied instead of Theorem 3.3, and again we refer to [22] for details.

A proof of Theorem 3.3 can be found in [35]. However, we wish to give an alternative proof which is inspired by [12], where the Banach contraction mapping theorem was proven from the fixed-point theorem for Scott continuous functions on complete partial orders. We will prove the Priess-Crampe and Ribenboim theorem using a form of Tarski's theorem, Theorem 3.5 below. For this purpose, we will impose the condition on the generalized ultrametric space (X, d, Γ) that Γ is of the form $\{2^{-\alpha} \mid \alpha \leq \gamma\}$ for some ordinal γ, ordered as above. Such a generalized ultrametric space will be called a gum with *ordinal distances*.

3.5 Theorem Let $(D, \leq)$ be a partially ordered set with bottom element $\perp$ such that each chain has a least upper bound, and let $f : D \to D$ be a monotonic function on D. Then f has a least fixed point.

For the proof of this well-known theorem, we define ordinal powers of f as follows. Let $f\uparrow 0 = \perp$; for each successor ordinal $\alpha + 1$, let $f\uparrow(\alpha + 1) = f(f\uparrow\alpha)$; for each limit ordinal α, let $f\uparrow\alpha = \sup\{f\uparrow\beta \mid \beta < \alpha\}$. The resulting chain of ordinal powers of f must become stationary at some stage, yielding a least fixed point of f.

The main technical tool which was employed in [12] is the space of formal balls associated with a given metric space. We will extend this notion to generalized ultrametrics.

Let (X, d, Γ) be a generalized ultrametric space and let $\mathcal{B}'X$ be the set of all pairs (x, α) with $x \in X$ and $\alpha \in \Gamma$. We define an equivalence relation $\sim$ on $\mathcal{B}'X$ by setting $(x_1, \alpha_1) \sim (x_2, \alpha_2)$ iff $\alpha_1 = \alpha_2$ and $d(x_1, x_2) \leq \alpha_1$. The quotient space $\mathcal{B}X = \mathcal{B}'X/\sim$ will be called the *space of formal balls associated with* (X, d, Γ), and carries an ordering $\sqsubseteq$ which is well-defined (on representatives of equivalence classes) by $(x, \alpha) \sqsubseteq (y, \beta)$ iff $d(x, y) \leq \alpha$ and $\beta \leq \alpha$. We denote the equivalence class of (x, α) by $[(x, \alpha)]$, and note of course that the use of the same symbol $\sqsubseteq$ between equivalence classes and their representatives should not cause confusion.

Informally, we think of each formal ball $[(x, \alpha)]$ in $\mathcal{B}X$ as standing for the ball $B_\alpha(x)$. The equivalence relation $\sim$ corresponds to the fact that $y \in B_\alpha(x)$ implies $B_\alpha(x) = B_\alpha(y)$, see Lemma 5.2. However, note that it is possible in general that $B_\alpha(x) = B_\beta(x)$, for some α and β, but that $\alpha \neq \beta$. So in this case we will still have $(x, \alpha) \not\sim (x, \beta)$. The ordering $\sqsubseteq$ is an abstract form of inverse containment for formal balls. More precisely, $(x, \alpha) \sqsubseteq (y, \beta)$ implies $B_\beta(y) \subseteq B_\alpha(x)$, but not vice versa in general.

3.6 Proposition The set $\mathcal{B}X$ is partially ordered by $\sqsubseteq$. Moreover, X is spherically complete if and only if every chain in $\mathcal{B}X$ has a least upper bound.

Proof: The proof is straightforward and we omit the details. ■

3.7 Proposition The function $\iota : X \to \mathcal{B}X : x \mapsto [(x, 0)]$ is injective and $\iota(X)$ is the set of all maximal elements of $\mathcal{B}X$.

Proof: Injectivity of ι follows from (Ui). The observation that the maximal elements of $\mathcal{B}X$ are exactly the elements of the form $[(x, 0)]$ completes the proof. ∎

Given a strictly contracting mapping f on a generalized ultrametric space (X, d, Γ) with ordinal distances, we define a function $\mathcal{B}f : \mathcal{B}X \to \mathcal{B}X$ by setting $\mathcal{B}f([(x, 2^{-\alpha})]) = \left[(f(x), 2^{-(\alpha+1)})\right]$, if $2^{-\alpha} \neq 0$, and setting $\mathcal{B}f([(x, 0)]) = [(f(x), 0)]$.

3.8 Proposition If f is strictly contracting, then $\mathcal{B}f$ is monotonic.

Proof: Let $(x, 2^{-\alpha}) \sqsubseteq (y, 2^{-\beta})$, so that $d(x, y) \leq 2^{-\alpha}$ and $\alpha \leq \beta$. Since $(x, 2^{-\alpha}) \sim (y, 2^{-\alpha})$, we only have to show that $d(f(x), f(y)) \leq 2^{-(\alpha+1)}$ (which holds since f is strictly contracting), that $\alpha + 1 \leq \beta + 1$ if $2^{-\beta+1} \in \Gamma$, that $\alpha + 1 \leq \beta$ if $2^{-\beta} = 0$ and $\alpha \neq \beta$, and that $\alpha \leq \beta$ if $2^{-\alpha} = 2^{-\beta} = 0$, all of which are easy to see. ∎

Proof of Theorem 3.3 for ordinal distances: Let (X, d, Γ) be a spherically complete generalized ultrametric space with ordinal distances, and let $f : X \to X$ be strictly contracting. Then $\mathcal{B}X$ is a partially ordered set such that every chain in $\mathcal{B}X$ has a least upper bound, and $\mathcal{B}f$ is a monotonic mapping on $\mathcal{B}X$. For $B_0 \in \mathcal{B}X$, we denote by $\uparrow B_0$ the upper cone of B_0, that is, the set of all $B \in \mathcal{B}X$ with $B_0 \sqsubseteq B$.

Let $x \in X$ be arbitrarily chosen, assuming without loss of generality that x is not a fixed point of f, and let α be an ordinal such that $d(x, f(x)) = 2^{-\alpha}$. Then $(x, 2^{-\alpha}) \sqsubseteq \left(f(x), 2^{-(\alpha+1)}\right)$ and, by monotonicity of $\mathcal{B}f$, we obtain that $\mathcal{B}f$ maps $\uparrow [(x, 2^{-\alpha})]$ into itself. Since $\uparrow [(x, 2^{-\alpha})]$ is a partially ordered set with bottom element $[(x, 2^{-\alpha})]$ and such that each chain in $\uparrow [(x, 2^{-\alpha})]$ has a least upper bound, we obtain that $\mathcal{B}f$ has a least fixed point in $\uparrow [(x, 2^{-\alpha})]$ which we will denote by B_0. It is clear by definition of $\mathcal{B}f$ that B_0 must be maximal in $\mathcal{B}X$, and hence is of the form $[(x_0, 0)]$. From $\mathcal{B}f[(x_0, 0)] = [(x_0, 0)]$, we obtain $f(x_0) = x_0$, so that x_0 is a fixed point of f.

Now assume that $y \neq x_0$ is another fixed point of f. Then $d(x_0, y) = d(f(x_0), f(y)) < d(x_0, y)$ since f is strictly contracting. This contradiction establishes that f has no fixed point other than x_0. ∎

Whilst the proof just given is a slight digression from the main theme of the paper, it does suggest the possibility of a domain-theoretic treatment of non-monotonic operators in logic programming, possibly related to the work of Rounds and Zhang in [39, 50, 51, 52]. We will return to this comment in Section 7. We also note that, on setting $\mathcal{B}f([(x, \alpha)]) = [(f(x), L(\alpha))]$, we can extend this proof from the case of ordinal distances to the slightly more general case that there exists a mapping $L : \Gamma \to \Gamma$ satisfying the following conditions: (i) L is monotonic, (ii) $L(0) = 0$, (iii) $L(\alpha)$ is the largest element of Γ strictly less than α for all $\alpha \neq 0$. Of course, this latter condition is as much a condition on Γ as it is a condition on L, and in particular it says that $L(\alpha) < \alpha$ for all $\alpha \neq 0$.

4 Dislocated Metrics and the Φ_ω^*-Accessible Programs

4.1 The Fixed-Point Theorem of Matthews

It will be convenient next to review the fixed-point theorem on dislocated metrics established by Matthews in [33]; it gives a result very similar in form to the Banach contraction mapping theorem.

4.1 Definition A sequence (x_n) in a dislocated metric space (X, ϱ) *converges with respect to* ϱ (or *in* ϱ) if there exists an $x \in X$ such that $\varrho(x_n, x)$ converges to 0 as $n \to \infty$. In this case, x is called the *d-limit* of (x_n).

It is easy to see that limits in dislocated metric spaces are unique.

4.2 Definition A sequence (x_n) in a dislocated metric space (X, ϱ) is called a *Cauchy sequence* if, for each $\varepsilon > 0$, there exists $n_0 \in \mathbb{N}$ such that for all $m, n \geq n_0$ we have $\varrho(x_m, x_n) < \varepsilon$. A dislocated metric space X is called *complete* if every Cauchy sequence in X converges. A function $f : X \to X$ is called a *contraction* if there exists $0 \leq \lambda < 1$ such that the inequality $\varrho(f(x), f(y)) \leq \lambda \varrho(x, y)$ holds for all $x, y \in X$.

4.3 Theorem Let (X, ϱ) be a complete dislocated metric space and let $f : X \to X$ be a contraction. Then f has a unique fixed point x_0 which can be obtained as the d-limit of the sequence $(f^n(x))_{n \in \mathbb{N}}$ for any $x \in X$.

Proofs of this theorem can be found in [33, 20]. Another proof, which is closer to the proof of the original Banach contraction mapping theorem, can be found in [24]. A third proof will be given in Section 4.3.

There are various ways of obtaining dislocated metrics from metrics, see [24]. In fact, the following result, which is Proposition 4.7 of [24], will be applied in Section 4.2.

4.4 Proposition Let (X, d) be an ultrametric space and let $u : X \to \mathbb{R}_0^+$ be a function. Then (X, ϱ) with

$$\varrho(x, y) = \max\{d(x, y), u(x), u(y)\}$$

is a d-ultrametric and $\varrho(x,x) = u(x)$ for all $x \in X$. If u is a continuous function on (X,d), then completeness of (X,d) implies completeness of (X,ϱ).

The function u is called a *weight*. In the definition of ϱ, we think of d as providing a basic, probably context-independent metric structure, while u encodes some particular knowledge about the specific problem at hand. In what follows, it may be helpful to think of $u(x)$ as measuring the extent to which x is *a priori not suitable* as a solution to a given problem. We will clarify this intuition in the next section.

4.2 An Application to the Φ_ω^*-Accessible Programs

In the following, P denotes a Φ_ω^*-accessible program which satisfies the defining conditions of such programs with respect to a model I and a level mapping l. Then (I_P, d), with d as given by Definition 3.1, is a complete ultrametric space.

Inspired by [16], we next define a function $f : I_P \to \mathbb{R}$ by setting $f(K) = 0$ if $K \subseteq I$ and, if $K \not\subseteq I$, by setting $f(K) = 2^{-n}$, where n is the smallest integer such that there is an atom $A \in B_P$ with $l(A) = n$, $K \models A$ and $I \not\models A$. Finally, we define $u : I_P \to \mathbb{R}$ by $u(K) = \max\{f(K'), d(K \setminus K', I \setminus I')\}$, where K', for any $K \in I_P$, denotes K restricted to the predicate symbols which are not in Neg_P^*, and we define $\varrho : I_P \times I_P \to \mathbb{R}$ by

$$\varrho(J,K) = \max\{d(J,K), u(J), u(K)\}.$$

We call ϱ the d-metric *associated with* P.

Proposition 4.4 yields that ϱ is a complete d-ultrametric on I_P provided that we are able to show that the function u, as given there, is continuous.

4.5 Lemma The function $u : I_P \to \mathbb{R}$ defined by $u(K) = \max\{f(K'), d(K \setminus K', I \setminus I')\}$ is continuous as a function from (I_P, d) to $\mathbb{R}$.

Proof: Let K_m be a sequence in I_P which converges in d to some $K \in I_P$. We need to show that $d(K \setminus K'_m, I \setminus I')$ converges to $d(K \setminus K', I \setminus I')$ and that $f(K'_m)$ converges to $f(K')$ as $m \to \infty$. Since (K_m) converges to K with respect to the metric d, it follows that for each $n \in \mathbb{N}$ there is $m_n \in \mathbb{N}$ such that K and K_m, for all $m \geq m_n$, agree on all atoms of level less than or equal to n. So, if $f(K) = 2^{-n_0}$, say, that means that K_m and K agree on all atoms of level less than or equal to n if $m \geq m_{n_0}$, and hence $f(K_m) = f(K)$ for all $m \geq m_{n_0}$. Also, if $d(K \setminus K', I \setminus I') = 2^{-n_0}$, say, then $d(K_m \setminus K'_m, I \setminus I')$=$d(K \setminus K', I \setminus I')$ for all $m \geq m_{n_0}$, as required. ■

As an example, we consider a rather simple instance of the program `game`, call it `game1`, which is as follows:

$$\begin{aligned} win(X) &\leftarrow move(X,Y), \neg win(Y) \\ move(a,b) &\leftarrow \\ move(a,c) &\leftarrow \\ move(b,d) &\leftarrow \end{aligned}$$

In order to clarify the role of the function u, we define program `game2` to consist of the clauses from `game1` together with the single clause

$$wins(X) \leftarrow win(X).$$

According to the analysis of `game` in [2], which is easily adapted to `game2`, we can construct a level mapping l and a model I such that `game2` is Φ_ω^*-accessible with respect to l and I. The definitions are as follows, where the graph G is given as the set $\{(a,b),(a,c),(b,d)\}$.

We define a function f' mapping $D = \{a,b,c,d\}$ into the set of natural numbers by $f'(c) = f'(d) = 0$, $f'(b) = 1$, $f'(a) = 2$, and we define the level mapping l by $l(move(x,y)) = f'(x)$ for all $(x,y) \in G$, $l(win(x)) = f'(x) + 1$ for $x \in D$, and $l(wins(x)) = f'(x) + 2$, thus

$$\begin{aligned} &l^{-1}(0) = \emptyset, \\ &l^{-1}(1) = \{move(b,d), win(c), win(d)\}, \\ &l^{-1}(2) = \{move(a,b), move(a,c), win(b)\} \cup \{wins(c), wins(d)\}, \\ &l^{-1}(3) = \{win(a)\} \cup \{wins(b)\}, \\ &l^{-1}(4) = \{wins(a)\}. \end{aligned}$$

We also define a function g' from D to $\{0,1\}$ by $g'(c) = g'(d) = 0$ and $g'(a) = g'(b) = 1$, and we also define the following interpretations

$$\begin{aligned} I' &= \{move(x,y) \mid (x,y) \in G\} \cup \{win(x) \mid g'(x) = 1\} \\ &= \{move(a,b), move(a,c), move(b,d), win(a), win(b)\} \qquad \text{and} \\ I &= I' \cup \{wins(x) \mid x \in D\} \\ &= \{move(a,b), move(a,c), move(b,d), win(a), win(b)\} \\ &\quad \cup \{wins(a), wins(b), wins(c), wins(d)\}. \end{aligned}$$

By [2], `game2` is Φ_ω^*-accessible with respect to l and I. This is also easily verified directly.

We now consider the construction of ϱ given earlier. We obtain $\mathrm{Neg}^*_{\mathtt{game2}} = \{win, move\}$ and `game2`$^-$ as the subprogram `game1`. Let us call an interpretation

J for `game2` (*a priori*) *suitable* if $J \subseteq I$ and J coincides with I on all atoms with predicate symbol in $\mathrm{Neg}^*_{\texttt{game2}}$. In the light of Proposition 4.11 below, this is equivalent to saying that $J \subseteq I$ and that $J \setminus J'$ is the unique supported model of `game1`. It then turns out that $u(K) = 0$ if and only if K is suitable. Furthermore, suitability of some interpretation K is a necessary condition for obtaining $\varrho(K, K) = 0$, which is in turn a necessary condition for K to be a fixed point of $T_{\texttt{game2}}$ provided the latter operator is a contraction (which is the case as we will see later). To summarize: a supported model of `game2` is always suitable. This is the precise sense in which $u(x)$, for given x, can be thought of as a quantitative measure of the extent to which x is a priori not suitable as a solution. The observations just given are valid in general, and we will return to this point at the end of the section. But first, we define suitability for Φ^*-accessible programs.

4.6 Definition Let P be a program which is Φ^*-accessible with respect to some level mapping l and some model I. An interpretation K of P is called (*a priori*) *suitable* if $K \subseteq I$ and K coincides with I on all atoms with predicate symbol in Neg^*_P.

The following result states that if an arbitrary sequence of interpretations converges with respect to ϱ, then its limit is a limit in Q and is always suitable.

4.7 Proposition Let P be a program which is Φ^*_ω-accessible with respect to a model I and a level mapping l, and let ϱ be its associated d-metric. If (J_n) is a sequence which converges in ϱ to some K, then (J_n) converges in the atomic topology on I_P, and the following two conditions hold.

(i) J_n restricted to Neg^*_P converges in Q, and its limit is I restricted to Neg^*_P.

(ii) J_n restricted to the complement of Neg^*_P converges in Q to some $J \subseteq I$.

Furthermore, the limit K of J_n is equal to $(I \setminus I') \cup J$.

Proof: It is easy to see that if $\varrho(J_n, K) < 2^{-k}$, then J_n and K agree on all atoms of level less than k which shows the first assertion. From convergence in Q of (J_n) to some K, it follows that $(J_n \setminus J'_n)$ and (J'_n) converge in Q to $K \setminus K'$ respectively K'. By definition of ϱ, we have $d(K \setminus K', I \setminus I') = 0$ which implies that $K \setminus K' = I \setminus I'$. From the same definition, we obtain $f(K) = 0$ and therefore $J = K' \subseteq I$ which completes the proof. ■

We return now to the study of the fixed-point theoretical aspects of Φ^*_ω-accessible programs. The proof of the following proposition carries over directly from the treatment of acceptable programs given in [16].

4.8 Proposition Let P be a Φ^*_ω-accessible program and let ϱ be defined as above. Then the associated immediate consequence operator T_P is a contraction on (I_P, ϱ).

By Theorem 4.3 we can now conclude the following result.

4.9 Theorem Each Φ^*_ω-accessible program has a unique supported model which can be obtained as the limit, in the atomic topology, of iterates of the single-step operator associated with the program.

Proof: Let P be Φ^*_ω-accessible. Then (I_P, ϱ) is a complete d-ultrametric space and T_P is a contraction relative to ϱ. By Theorem 4.3, T_P has a unique fixed point which is the unique supported model of P. This fixed point can be obtained as stated by Proposition 4.7. ∎

In the remainder of this section, let P be a program which is Φ^*_ω-accessible with respect to some level mapping l and some interpretation I, and let M_P be the unique supported model of P. We now formally prove that M_P is a priori suitable.

4.10 Proposition If J is an interpretation with $\varrho(J, J) = 0$, then J is suitable. In particular, M_P is suitable.

Proof: From $\varrho(J, J) = 0$ we obtain $u(J) = 0$, and from the definition of u the assertion follows. Since $T_P(M_P) = M_P$, and since T_P is a contraction, we obtain $\varrho(M_P, M_P) = 0$. ∎

The construction of the dislocated metric ϱ depends on the interpretation I with respect to which P is Φ^*_ω-accessible. Determining such an interpretation from a given program is an undecidable task, and cannot be automated. So we can only hope to give some guidance, in the form of necessary conditions, which can help in determining it. Such necessary conditions on interpretations and on level mappings can be found in [20, Section 5.2]. We next prove a result which is important in this respect. It will guide our search for an interpretation with respect to which the program P_f from Section 2.2 is Φ^*-accessible, and this point will be taken up at the end of Section 5.2.

4.11 Proposition The following hold.

(i) For every suitable J with $M_P \subseteq J \subseteq I$, we have that P is Φ^*_ω-accessible with respect to J and l.

(ii) M_P is the intersection of all models K with respect to which P is Φ^*_ω-accessible.

(iii) Every interpretation K with respect to which P is Φ_ω^*-accessible contains M_P and coincides with M_P on all atoms with predicate symbol in Neg_P^*.

Proof: Since J is suitable, we obtain that J coincides with I on all atoms with predicate symbol in Neg_P^*. Thus, for all clauses in P^-, the defining condition for Φ_ω^*-accessibility is satisfied. The remaining clauses form a definite program. Now let $A \leftarrow B_1, \ldots, B_n$ be a ground instance of such a clause. If $B_i \in I$ for all $i = 1, \ldots, n$, then $l(B_i) < l(A)$ for all i and the defining condition holds also with respect to J. If there is some i with $B_i \notin I$ and $l(B_i) < l(A)$, then $B_i \notin J$ since J is suitable, and again the defining condition holds.

This much proves (i), from which it follows that P is also Φ_ω^*-accessible with respect to M_P and l. Hence, $M_P \supseteq \bigcap \mathcal{I}$, where $\mathcal{I}$ stands for the set of all interpretations with respect to which P is Φ_ω^*-accessible. Now let $K \in \mathcal{I}$. From suitability of M_P, we obtain both the inclusion $M_P \subseteq K$ and the coincidence of M_P with K on all atoms with predicate symbol in Neg_P^*, and this establishes (iii). Finally, the inclusion $M_P \subseteq K$ implies $M_P \subseteq \bigcap \mathcal{I}$, proving the equality stated in (ii). ■

4.3 Dislocated Metrics and Metrics

In this subsection, we consider the relationship between the theorem of Matthews, Theorem 4.3 and the Banach contraction mapping theorem.

4.12 Proposition Let (X, ϱ) be a dislocated metric space and define $d : X \times X \to \mathbb{R}$ by setting $d(x, y) = \varrho(x, y)$ for $x \neq y$ and setting $d(x, x) = 0$ for all $x \in X$. Then d is a metric.

Proof: We obviously have $d(x, x) = 0$ for all $x \in X$. If $d(x, y) = 0$, then either $x = y$ or $\varrho(x, y) = 0$, and from the latter we also obtain $x = y$. Symmetry is clear. We want to show that $d(x, y) \leq d(x, z) + d(z, y)$ for all $x, y, z \in X$. If $d(x, z) = \varrho(x, z)$ and $d(z, y) = \varrho(z, y)$, then the inequality is clear. If $d(x, z) = 0$, then $x = z$ and the inequality reduces to $d(x, y) \leq d(x, y)$ which holds. If $d(z, y) = 0$, then $z = y$ and the inequality reduces to $d(x, y) \leq d(x, y)$ which also holds. ■

We call d, as defined in Proposition 4.12, the metric *associated* with the d-metric ϱ. We note that if $x \in X$ has self-distance $\varrho(x, x) \neq 0$, then for each $y \neq x$ we obtain $d(x, y) = \varrho(x, y) \geq \frac{1}{2}\varrho(x, x)$ by the triangle inequality. Consequently, each $x \in X$ with $\varrho(x, x) \neq 0$ is an isolated point with respect to the metric d associated with ϱ, that is, for each such x there exists $\varepsilon > 0$ such that the ball with centre x and radius ε contains only the single point x.

4.13 Proposition Let (X, ϱ) be a dislocated metric space and let d denote the metric associated with ϱ. If the metric d is complete, then so is ϱ. If f is a contraction relative to ϱ, then f is also a contraction relative to d.

Proof: If (x_n) is a Cauchy sequence in ϱ, then for all $\varepsilon > 0$ there exists n_0 such that $\varrho(x_k, x_m) < \varepsilon$ for all $k, m \geq n_0$. Consequently, we also obtain $d(x_k, x_m) < \varepsilon$ for all $k, m \geq n_0$ and, since d is complete, the sequence (x_n) converges in d to some x. Thus, $d(x_n, x) \to 0$ as $n \to \infty$. It remains to show that $\varrho(x_n, x) \to 0$ as $n \to \infty$, and we consider two cases.
(1) Assume that the sequence (x_n) is such that there exists n_0 with $x_m \neq x$ for all $m \geq n_0$. Then $\varrho(x_m, x) = d(x_m, x)$ for all $m \geq n_0$, that is, $\varrho(x_m, x) \to 0$, and hence $\varrho(x_n, x) \to 0$.
(2) Assume that there exist infinitely many $n_k \in \mathbb{N}$ such that $x_{n_k} = x$. Since (x_n) is a Cauchy sequence with respect to ϱ, we obtain $\varrho(x_{n_k}, x) < \varepsilon$ for all $\varepsilon > 0$, that is, $\varrho(x, x) = 0$. Hence, $\varrho(x_n, x) = d(x_n, x)$ for all $n \in \mathbb{N}$, as required.

Let $x, y \in X$ and assume $\varrho(f(x), f(y)) \leq \lambda \varrho(x, y)$ for some $0 \leq \lambda < 1$. If $f(x) = f(y)$, then $d(f(x), f(y)) = 0$, hence $d(f(x), f(y)) \leq \lambda d(x, y)$. If $f(x) \neq f(y)$, then $x \neq y$ and so $d(f(x), f(y)) = \varrho(f(x), f(y)) \leq \lambda \varrho(x, y) = \lambda d(x, y)$, as required. ■

4.14 Proposition Let (X, ϱ) be a complete dislocated metric space and let d denote the metric associated with ϱ. Then the metric d is complete. However, a function f can be a contraction relative to d and not be a contraction relative to ϱ.

Proof: Let (x_n) be a Cauchy sequence in d. If (x_n) eventually becomes constant, then it obviously converges in d. So assume that (x_n) does not eventually become constant. Then in that case we note that (x_n) contains infinitely many distinct points, otherwise it would not be a Cauchy sequence. Now define a subsequence (y_n) of (x_n) by removing multiple occurrences of points in (x_n): for each $n \in \mathbb{N}$, let $y_n = x_k$, where k is minimal with the property that for all $m < n$ we have $y_m \neq x_k$. Since (y_n) is a subsequence of the Cauchy sequence (x_n), we obtain that (y_n) is also a Cauchy sequence. Now, for any two elements y, z in the sequence (y_n), we have that $d(y, z) = \varrho(y, z)$ by definition of d, and hence (y_n) converges in ϱ to some $y_\omega \in X$. Hence, (y_n) also converges in d to y_ω. We show next that (x_n) converges to y_ω. Let $\varepsilon > 0$ be arbitrarily chosen. Since (x_n) is a Cauchy sequence, there exists an index n_1 such that $d(x_k, x_m) < \frac{\varepsilon}{2}$ for all $k, m \geq n_1$. Since (y_n) converges to y_ω, we also know that there is an index n_2 with $y_{n_2} = x_{n_3}$ for some index n_3 such that $n_3 \geq n_1$ and that $d(y_{n_2}, y_\omega) < \frac{\varepsilon}{2}$. For all x_n with $n \geq n_3$, we then obtain $d(x_n, y_\omega) \leq d(x_n, x_{n_3}) + d(x_{n_3}, y_\omega) < \varepsilon$, as required.

Let $X = \{0, 1\}$, and define a mapping $f : X \to X$ by $f(x) = 0$ for $x \in X$. Let ϱ be constant and equal to 1. Then ϱ is a complete d-metric and f is a contraction relative to d. However, $\varrho(f(0), f(1)) = \varrho(0, 0)$, and so f is not a contraction relative to ϱ. ■

We can now prove Theorem 4.3 by using the Banach contraction mapping theorem as follows. Let ϱ be a complete d-metric and let f be a contraction relative to ϱ. Let d denote the metric associated with ϱ. Then d is a complete metric and f is a contraction relative to d. So, f has a unique fixed point by the Banach contraction mapping theorem.

5 Dislocated Generalized Ultrametrics and the Φ^*-Accessible Programs

In this section, we extend Theorem 3.3 by allowing non-zero self-distances. The results we establish will then be applied to discuss the fixed-point theory of the class of Φ^*-accessible programs (defined in Section 2) which we introduced in [21] in the context of three-valued logical operators.

5.1 A Generalized Priess-Crampe & Ribenboim Fixed-Point Theorem

5.1 Definition Let (X, d, Γ) be a dislocated generalized ultrametric space. A function $f : X \to X$ is called *strictly contracting* if $d(f(x), f(y)) < d(x, y)$ for all $x, y \in X$ with $x \neq y$.

We will need the following observations, which are well-known for ultrametric spaces.

5.2 Lemma Let (X, d, Γ) be a dislocated generalized ultrametric space. For $\alpha, \beta \in \Gamma$ and $x, y \in X$, the following statements hold.

(1) If $\alpha \leq \beta$ and $B_\alpha(x) \cap B_\beta(y) \neq \emptyset$, then $B_\alpha(x) \subseteq B_\beta(y)$.

(2) If $B_\alpha(x) \cap B_\alpha(y) \neq \emptyset$, then $B_\alpha(x) = B_\alpha(y)$.

(3) $B_{d(x,y)}(x) = B_{d(x,y)}(y)$.

Proof: Let $a \in B_\alpha(x)$ and let $b \in B_\alpha(x) \cap B_\beta(y)$. Then $d(a, x) \leq \alpha$ and $d(b, x) \leq \alpha$, so by (Uiv) $d(a, b) \leq \alpha \leq \beta$. Since $d(b, y) \leq \beta$, we obtain from (Uiv) again that $d(a, y) \leq \beta$, so that $a \in B_\beta(y)$, which proves the first statement. The second follows by symmetry, and the third by replacing α by $d(x, y)$. ■

The following theorem reconciles the theorem of Matthews (Theorem 4.3) and the Priess-Crampe & Ribenboim theorem (Theorem 3.3) for ordinal distances. Although the proof of the latter theorem given in [35] carries over directly to our more general setting, see [26], we give a new proof which is constructive in that it yields a way of obtaining the fixed point as a kind of limit. The original proof shows existence only, whereas the new proof is in the spirit of [31].

5.3 Theorem Let (X, d, Γ) be a spherically complete dislocated generalized ultrametric space with ordinal distances, so that $\Gamma = \{2^{-\alpha} \mid \alpha \leq \gamma\}$ for some ordinal γ (as usual we order Γ by $2^{-\alpha} < 2^{-\beta}$ iff $\beta < \alpha$, and denote $2^{-\gamma}$ by 0). If $f : X \to X$ is any strictly contracting function on X, then f has a unique fixed point.

Proof: Let $x \in X$. Then $f(x) \in f(X)$ and $d(f(x), x) \leq 2^{-0}$, since 2^{-0} is the maximum distance possible between any two points in X. Now, $d(f(f(x)), f(x)) \leq 2^{-1} \leq 2^{-0}$ since f is strictly contracting, and by (Uiv) it follows that $d(f^2(x), x) \leq 2^{-0}$. By the same argument, we obtain $d(f^3(x), f^2(x)) \leq 2^{-2} \leq 2^{-1}$ and therefore $d(f^3(x), f(x)) \leq 2^{-1}$. In fact, an easy induction argument along these lines shows that $d(f^{n+1}(x), f^m(x)) \leq 2^{-m}$ for $m \leq n$. Again by (Uiv), we obtain that the sequence of balls of the form $B_{2^{-n}}(f^n(x))$ is a descending chain (with respect to set-inclusion) if n is increasing, and therefore has non-zero intersection B_ω since X is spherically complete. We therefore conclude that there is $x_\omega \in B_\omega$ with $d(x_\omega, f^n(x)) \leq 2^{-n}$ for each $n \in \mathbb{N}$.

For each $n \in \mathbb{N}$ we argue as follows. Since $d(f(x_\omega), f^{n+1}(x)) < d(x_\omega, f^n(x)) \leq 2^{-n}$ and $d(x_\omega, f^{n+1}(x)) \leq 2^{-(n+1)} \leq 2^{-n}$, we obtain $d(f(x_\omega), x_\omega) \leq 2^{-n}$ by (Uiv). Since this is the case for all $n \in \mathbb{N}$, we obtain $d(f(x_\omega), x_\omega) \leq 2^{-\omega}$.

It is straightforward to cast the observations above into a transfinite induction argument, and we obtain the following construction: Choose $x \in X$ arbitrarily. For each ordinal $\alpha \leq \gamma$, we define $f^\alpha(x)$ as follows. If α is a successor ordinal, then $f^\alpha(x) = f(f^{\alpha-1})(x)$. If α is a limit ordinal, then we choose $f^\alpha(x)$ as some x_α which has the property that $d(x_\alpha, f^\beta(x)) \leq 2^{-\beta}$, and we note that the existence of such an x_α is guaranteed by spherical completeness of X.

The resulting transfinite sequence $f^\alpha(x)$ has the property that $d(f^{\alpha+1}, f^\alpha) \leq 2^{-\alpha}$ for all $\alpha \leq \gamma$. Consequently, $d(f^{\gamma+1}(x), f^\gamma(x)) = 2^{-\gamma} = 0$, and therefore $f^\gamma(x)$ must be a fixed point of f.

Finally, $x_\gamma = f^\gamma(x)$ can be the only fixed point of f. To see this, suppose $y \neq x_\gamma$ is another fixed point of f. Then we obtain $f(y, x_\gamma) < f(y, x_\gamma)$ from the fact that f is strictly contracting, which is impossible. ■

5.4 Proposition Let (X, d, Γ) be a generalized ultrametric space, where Γ is a complete lattice, and let $u : X \to \mathbb{R}_0^+$ be a function. Then (X, ϱ, Γ) with

$$\varrho(x, y) = \sup\{d(x, y), u(x), u(y)\}$$

is a dislocated generalized ultrametric space, and $\varrho(x, x) = u(x)$ for all $x \in X$.

Proof: (Ui) and (Uiii) are trivial, and (Uiv) is proved in the same way that Proposition 4.4 is proved. ■

5.2 Application to the Φ^*-Accessible Programs

We intend now to apply the results above to the Φ^*-accessible programs. We do this, in effect, by merging the lines of thinking employed in Sections 3 and 4.2.

In the following, P denotes a Φ^*-accessible program which satisfies the defining conditions for such programs with respect to a model I and a level mapping $l : B_P \to \gamma$. Recall that the space (I_P, d), as given by Definition 3.3, is a generalized ultrametric space.

Following Section 4.2, we define a function f on I_P by setting $f(K) = 0$ if $K \subseteq I$ and, if $K \not\subseteq I$, by setting $f(K) = 2^{-\alpha}$, where α is the smallest ordinal such that there is an atom $A \in B_P$ with $l(A) = \alpha$, $K \models A$ and $I \not\models A$. Finally, we define u on I_P by $u(K) = \max\{f(K'), d(K \setminus K', I \setminus I')\}$, where K', for any $K \in I_P$, denotes K restricted to the predicate symbols which are not in Neg_P^*, and we define ϱ on $I_P \times I_P$ by

$$\varrho(J, K) = \sup\{d(J, K), u(J), u(K)\} = \max\{d(J, K), u(J), u(K)\}.$$

5.5 Proposition The space (I_P, ϱ) is a spherically complete dislocated generalized ultrametric space.

Proof: (Ui), (Uiii) and (Uiv) follow from Proposition 5.4. For spherical completeness, let $(\mathcal{B}_\alpha)$ be a chain of balls in X with midpoints I_α. Let I be the set of all atoms which are *eventually* in I_α, that is, the set of all $A \in B_P$ such that there exists some β with $A \in I_\alpha$ for all $\alpha \geq \beta$. It is easy to see that for each ball $B_{2^{-\alpha}}$ in the chain, we have $d(I_\alpha, I) \leq 2^{-\alpha}$, and hence I is in the intersection of the chain. ■

The proof of the next proposition is analogous to that of [16, Lemma 7.1 and Proposition 7.1], and the details can be found in [26].

5.6 Proposition Let P be Φ^*-accessible with respect to a level mapping l and a model I. Then we have $\varrho(T_P(J), T_P(K)) < \varrho(J, K)$ for all $J, K \in I_P$ with $J \neq K$.

5.7 Theorem Let P be a Φ^*-accessible program. Then P has a unique supported model.

Proof: By Proposition 5.6, T_P is strictly contracting with respect to ϱ, which in turn is a spherically complete dislocated generalized ultrametric. By Theorem

5.3, the operator T_P must have a unique fixed point and hence P has a unique supported model. ■

Using the proof of Theorem 5.3 above, we can in fact obtain the unique model by constructing the sequence $f^\beta(\emptyset)$ as shown in that proof, where f denotes the operator T_P. It remains to determine how to obtain $f^\beta(\emptyset)$ in the case that β is a limit ordinal. To this end, we employ the construction from the proof of Proposition 5.5, that is, we set $f^\beta(\emptyset)$ to be the set of all $A \in B_P$ which are eventually in $(f^\alpha(\emptyset))_{\alpha<\beta}$. The transfinite sequence thereby obtained has the property that it converges in the atomic topology to the unique fixed point of f.

As an example, we will now return, as earlier promised, to the analysis of the program P_f from Section 2.2. To do this, we will first determine a level mapping l and a model I such that P is Φ^*-accessible with respect to l and I. Since there is no algorithm for computing I, we give a heuristic argument for constructing I and l, and then in fact *prove* Φ^*_ω-accessibility of P_f with respect to I and l.

We assume that the unique supported model M_{P_g} of the Φ^*_ω-accessible program P_g contains exactly those atoms of the form $p_g(x_1, \ldots, x_n, y, u)$ for which $g(x_1, \ldots, x_n, y)$ is not undefined and equal to u, and that the collection of all $l(A)$ for which A has predicate symbol p_g is bounded by some limit ordinal $\alpha > \omega$. It is also a reasonable assumption that the only constant symbol occurring in P_g is 0, and that the only function symbol occurring in P_g is s.

From the insights obtained by Proposition 4.11, in the case of Φ^*_ω-accessible programs, we start looking for a model I which is the unique supported model of P_f^- when restricted to the predicate symbols in $\mathrm{Neg}^*_{P_f}$. We also know that P_g is Φ^*_ω-accessible with respect to M_{P_g} and a suitable level mapping l_g. So, it is reasonable to work under the assumption that I, restricted to the atoms with predicate symbols occurring in P_g, coincides with M_{P_g}, and that l, restricted to these atoms, coincides with l_g.

We next turn to the predicate symbol *lt*. The subprogram consisting of the two clauses which define *lt* is acyclic with respect to the level mapping l_{lt} which maps each atom $lt(s^m(0), s^n(0))$ to m. It has unique supported model $M_{lt} = \{lt(s^m(0), s^n(0)) \mid m < n\}$, and we assume now that I coincides with M_{lt} on all atoms which contain *lt*.

The difficult task which remains is to determine what values I should have on atoms with predicate symbol r. Obviously, I should contain all atoms of the form $r(s^m(0), s^n(0), 0, s^n(0))$, since I must be a model of P_f. Which of the atoms of the form $r(s^m(0), s^n(0), s^{k+1}(0), s^j(0))$ must be true under I? At most those for which $n < j$, since otherwise the atom $lt(s^n(0), s^j(0))$ occurring in the corresponding body is false in I. From Proposition 4.11 again, we can borrow the intuition that it is not unreasonable to try with an I which is larger than the unique supported model of P_f, so we try the following: let I contain

$M_r = \{r(s^n(0), s^m(0), s^k(0), s^j(0)) \mid m \leq j\}$. We now turn to the level mapping. If we take one of the atoms of the form $r(s^m(0), s^n(0), s^{k+1}(0), s^j(0))$, with $n < j$, as head of the recursive clause in the definition of r, then we note that the recursive call is made with the atom $r(\cdot, s^{n+1}(0), \cdot, s^j(0))$, where $\cdot$ denotes some term, and we have $n + 1 \leq j$. So the *difference* between the numbers represented by the second and fourth argument decreases, and we define $l(r(s^n(0), s^m(0), s^k(0), s^j(0))) = \alpha + j - m$ for $j \geq m$ and $= \alpha$ otherwise. Finally, we add to I the set M_{p_f} of all atoms containing p_f, and set the level of all these atoms to $\alpha + \omega$.

Thus, in summary:

$$\begin{aligned} I &= M_{P_g} \cup M_{lt} \cup M_r \cup M_{p_f} \\ &= M_{P_g} \cup M_{p_f} \cup \{lt(s^n(0), s^m(0)) \mid n < m\} \\ &\quad \cup \{r(s^n(0), s^m(0), s^k(0), s^j(0)) \mid m \leq j\} \\ l(lt(s^n(0), s^m(0)) &= n \\ l(r(s^n(0), s^m(0), s^k(0), s^j(0))) &= \begin{cases} \alpha + j - m, & \text{if } j \geq m, \\ \alpha, & \text{otherwise} \end{cases} \\ l(p_f(s^n(0), s^m(0))) &= \alpha + \omega \end{aligned}$$

It is easy to verify now that P_f is Φ^*-accessible with respect to I and l as defined above.

5.3 Generalized Ultrametrics and Dislocated Generalized Ultrametrics

We want next to investigate the relationship between Theorem 5.3 and Theorem 3.3.

5.8 Proposition Let (X, ϱ, Γ) be a dislocated generalized ultrametric space and define $d : X \times X \to \Gamma$ by $d(x, y) = \varrho(x, y)$ for $x \neq y$ and $d(x, x) = 0$ for all $x \in X$. Then d is a generalized ultrametric.

Proof: The proof is straightforward and parallels the proof of Proposition 4.12. ■

We call d, as defined in Proposition 5.8, the generalized ultrametric *associated* with ϱ.

5.9 Proposition Let (X, ϱ, Γ) be a dislocated generalized ultrametric space and let d denote the generalized ultrametric associated with ϱ. If d is spherically complete, then ϱ is spherically complete. If f is strictly contracting relative to ϱ, then f is also strictly contracting relative to d.

Proof: Let $\mathcal{B}$ be a chain of non-empty balls in ϱ, thus each $B \in \mathcal{B}$ is of the form $B = \{x \in X \mid \varrho(x, m_B) \leq \alpha_B\}$ for some $m_B \in X$ and $\alpha_B \in \Gamma$. For each ball B in $\mathcal{B}$, we define $B' = B \cup \{m_B\}$. The collection $\mathcal{B}'$ of all the B' is a chain of balls in (X, d) and has non-empty intersection by assumption; let $x \in \bigcap \mathcal{B}'$. Assume that $x \notin \bigcap \mathcal{B}$. Then there must be a ball $B \in \mathcal{B}$ such that $x \in B'$ but $x \notin B$. Since B is non-empty, we also conclude that there is some $y \in B$ with $x \neq y$ and $\varrho(x, y) < \varrho(x, x)$. But by (Uiv) we know that $\varrho(x, x) \leq \varrho(x, y)$. This contradiction shows that our assumption $x \notin \bigcap \mathcal{B}$ is incorrect, and hence ϱ is spherically complete, as required.

Let $x, y \in X$ with $x \neq y$, and assume $\varrho(f(x), f(y)) < \varrho(x, y)$. If $f(x) = f(y)$, then $d(f(x), f(y)) = 0$, hence $d(f(x), f(y)) < d(x, y)$. If $f(x) \neq f(y)$, then $x \neq y$ and so $d(f(x), f(y)) = \varrho(f(x), f(y)) < \varrho(x, y) = d(x, y)$, as required. ■

5.10 Proposition Let (X, ϱ) be a spherically complete dislocated generalized ultrametric space and let d denote the generalized ultrametric associated with ϱ. Then d is spherically complete. However, a function f can be a contraction relative to d and not be a contraction relative to ϱ.

Proof: Let $\mathcal{B}$ be a chain of balls in d. For each ball $\{x \mid d(x, x_m) \leq \gamma\}$, we define a corresponding ball $\{x \mid \varrho(x, x_m) \leq \gamma\}$, where x_m is a midpoint of the ball in question. Let the corresponding chain of balls be denoted by $\mathcal{B}'$. Then $\bigcap \mathcal{B}' \neq \emptyset$. Since $\bigcap \mathcal{B}' \subseteq \bigcap \mathcal{B}$, we obtain $\bigcap \mathcal{B} \neq \emptyset$.

Let $X = \{0, 1\}$ and define a mapping $f : X \to X$ by $f(x) = 0$ for $x \in X$. Let ϱ be constant and equal to 1. Then $(X, \varrho, \{0, 1\})$, where $0 < 1$, is spherically complete, and f is strictly contracting relative to d. However, $\varrho(f(0), f(1)) = \varrho(0, 0)$, and so f is not strictly contracting relative to ϱ. ■

We can now use Theorem 3.3 to give an easy proof of a more general version of Theorem 5.3 which was already obtained by us using different methods in [26].

5.11 Theorem Let (X, ϱ, Γ) be a spherically complete dislocated ultrametric space and let $f : X \to X$ be strictly contracting on X. Then f has a unique fixed point.

Proof: Using Proposition 5.8, we obtain a generalized ultrametric d which is spherically complete by Proposition 5.10. By Proposition 5.9, the function f is strictly contracting relative to d. Hence, by Theorem 3.3, f has a unique fixed point. ■

6 Φ-Accessible Programs

We intend finally to consider the Φ-accessible programs using the same sort of methods as we used in Section 5.2. The approach does not generalize without modifications to Φ-accessible programs as can be seen from the following example program P.

$$\begin{aligned} p(s^2(x)) &\leftarrow p(x) \\ p(0) &\leftarrow \\ p(s^4(0)) &\leftarrow p(s^5(0)) \\ p(s^2(0)) &\leftarrow p(s^3(0)) \end{aligned}$$

This program is Φ-accessible (even definite Φ_ω-accessible) with respect to the model $B_P = \{s^n(0) \mid n \in \mathbb{N}\}$ and the level mapping $l : B_P \to \mathbb{N} : p(s^n(0)) = n$. Using the dislocated ultrametric ϱ from Section 5.2, we obtain for $K = \{s^5(0)\}$ and $J = \{s^3(0)\}$ that $\varrho(K, J) = 2^{-3}$ and $\varrho(T_P(K), T_P(J)) = 2^{-2}$, so T_P is not a contraction relative to ϱ.

We will modify the methods used in Section 5.2 by means of the following result.

6.1 Proposition Let (X, d, Γ) be a generalized ultrametric space with ordinal distances and define the function ϱ by

$$\varrho : X \times X \to \Gamma : (J, K) \mapsto \max\{d(J, I), d(I, K)\},$$

where I denotes any fixed element of X. Then (X, ϱ, Γ) is a dislocated generalized ultrametric space. Furthermore, if (X, d) is spherically complete, then so is (X, ϱ).

Proof: Clearly, ϱ is a d-gum. For spherical completeness, note that every non-empty ball in (X, ϱ) contains I, which suffices. ■

6.2 Proposition Let P be Φ-accessible with respect to a model I and level mapping l. Then T_P is strictly contracting with respect to (I_P, ϱ).

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 $l(A) \leq \alpha$. Then there must be a clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) such that $J \models L_1 \wedge \cdots \wedge L_n$. Since I and J agree on all ground atoms of level less than α, condition (ii) of Definition 2.2 cannot hold, because if $I \not\models L_i$ with $l(A) > l(L_i)$, then $J \not\models L_i$ and consequently $J \not\models L_1 \wedge \cdots \wedge L_n$, which is a contradiction. Therefore, condition (i) of Definition 2.2 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 $A \leftarrow L_1, \ldots, L_n$ in ground(P) such that $I \models L_1 \wedge \cdots \wedge L_n$. Thus, condition (i) of Definition 2.2 must hold, and so we can assume that $A \leftarrow L_1, \ldots, L_n$ also satisfies $l(A) > l(L_i)$ for $i = 1, \ldots, n$. Since I and J agree on all ground atoms of level less than α, we have $J \models L_1 \wedge \cdots \wedge L_n$ and hence $A \in T_P(J)$ as required. ■

Applying Theorem 5.3 now yields a unique fixed point M of the operator T_P, that is, a unique supported model for P. The proof of Theorem 5.3 furthermore yields that there must be an ordinal α such that $\varrho(M, M) = 0$. Since the only point of X which has non-zero distance from itself is I, we conclude that $I = M$ is the unique supported model of P. This is somewhat unfortunate since I was needed in order to construct ϱ. However, using the proof of Theorem 5.3 again, we can also conclude that transfinite iterates of T_P starting at an arbitrary interpretation J converge in Q to the unique fixed point. This latter result is stronger than that obtained in [21], using different methods, where this same fact was established only for $J = \emptyset$.

7 Related and Further Work

One of the early landmark results in the theory of logic programming, the Kowalski-van Emden Theorem, has topological content: the single-step operator for definite programs is continuous with respect to the Scott topology. For logic programming with negation, certain natural semantic operators appearing in the literature fail to have this property, despite the fact that their associated fixed-point semantics, given by the fixed points of these operators, is computationally meaningful. From this point of view, logic programming is unusual compared with other major programming paradigms, where most of the operators which arise are Scott continuous. In fact, the quest for a clear declarative meaning for negation in logic programming is still ongoing despite a very considerable amount of research having been undertaken on this problem.

The use of topology in the analysis of negation in logic programming was initiated by Batarekh and Subrahmanian who defined the query topology in [6, 7, 8]. Their work was considerably simplified and extended by Seda in [41]. Perhaps the best-known paper in this area is Fitting's paper [16] in which the use of metrics is proposed in the context of logic programming semantics. In [31], Khamsi, Kreinovich and Misane used metrics and a version of the Banach contraction mapping theorem for multivalued mappings, due to S.B. Nadler, to study answer set semantics for disjunctive logic programs. Stimulated by recent developments in domain theory, Seda in [42] investigated quasi-metrics and the corresponding fixed-point theorem on quasi-metric spaces due to Smyth [45] and Rutten [40], thus reconciling the use of metric spaces and order within logic programming. A quite recent development in this respect is due to Priess-Crampe and Ribenboim [35] who introduced generalized ultrametrics to logic programming and proposed to examine the applicability of their fixed-point theorems on these spaces, see [22] for an overview.

The generalized-metric approach presented in this paper offers a technique for studying fixed-point semantics of non-monotonic semantic operators. It can be expected that results similar to those presented herein can be obtained for semantics which are more refined than the Clark completion semantics. In particular, we see a promising candidate in semantics related to the **s**-semantics, see [9]. Such investigations may eventually lead to a practically useful declarative understanding of negation in logic programming.

Another aspect of the work presented in this paper concerns investigations into continuous models of computation. This encompasses the task of embedding semantic operators into Euclidean space, which carries many natural metrics. Of the many convincing arguments given by Blair et al. [4] for the virtue of such investigations, we point to one: the study of relationships between logic proramming and artificial neural networks. Indeed these two paradigms differ very much in their strengths and weaknesses, and it would be highly desirable to merge them. Some of the results in this area employ topological notions: Hölldobler, Störr and Kalinke [29] make use of the metric d from Section 3 and the Banach contraction mapping theorem; [28] uses the atomic topology. The paper [30] provides an overview of current open challenges concerning logic and neural networks.

Yet another prospect for further work lies in the intersection of topology and programming language semantics, that is, Domain Theory. Motivated by the fact that domains and logic are strongly related, see [49], Rounds and Zhang have extensively investigated domain-theoretical foundations of logic programming. In closing, we mention two lines of their work and how relationships between our work and theirs may possibly be established, as follows.

In [50, 51, 39], Rounds and Zhang develop a domain-theoretic perspective of default logic, using power domains, which they call *power defaults.* Default logic, due to Reiter [38], also motivated the development of the stable model semantics of Gelfond and Lifschitz [18], which is a refinement of the Clark completion semantics. Fages [13, 14] has studied interesting relationships between the Clark completion semantics and the stable semantics, and this may allow one to carry over some "metric" techniques to the stable model semantics. In order to establish connections with the work of Rounds and Zhang, it will be necessary to understand the exact relationship between the model-theoretic semantics of power defaults and the stable model semantics, leading perhaps to the transfer of metric methods to power defaults by means of the work of Fages. Such a transfer may help in understanding both logic programming and default reasoning.

In [39, 52], Rounds and Zhang have introduced a domain-theoretic framework for the study of semantical aspects of logic programming, including an abstract resolution rule. The main gap between this work and the Clark completion semantics (and practical logic programming) in our opinion lies in the treatment of negation, which is not taken to be finite failure in [52]. In order to establish a cross-transfer of methods, it seems to be necessary to first study the semantics of Rounds and Zhang in the context of negation as failure. Again, a possible long-term prospect lies in a clean domain-theoretic treatment of the declarative semantics of negation in logic programming.

8 Conclusions

We have presented a unified framework for the fixed-point analysis of normal logic programs with respect to the Clark completion semantics which, as is well-known, is strongly related procedurally to negation as finite failure. This was achieved by casting spaces of interpretations into generalized metric spaces in such a way that various fixed-point theorems could be applied. The classes of programs which we investigated were all contained in the rather general class of programs which have a total Kripke-Kleene semantics, and encompass classes which are important from the point of view of termination analysis, and classes which are sufficiently expressive for every partial recursive function to be implemented within them. Each of the programs studied has a unique supported model which can be obtained as a limit in the atomic topology of iterates of the immediate consequence operator, starting from any arbitrarily chosen interpretation. For small classes of programs such as the acyclic or locally hierarchical programs, no semantic knowledge about the programs is needed in order to construct the generalized metric used here. The more we relax the requirements on the programs, the more knowledge is built into the construction of the generalized metric. At each stage in the development, we have investigated the generalized metric spaces which were appropriate at that stage, and have given new proofs of some of the fixed-point theorems we applied. Finally, we have presented three possible lines of investigation which are related to our work, and which can be expected to further our knowledge of theoretical and practical aspects of logic programming.

References

[1] K.R. Apt, H.A. Blair and A. Walker, *Towards a Theory of Declarative Knowledge.* In: J. Minker (Ed.), Foundations of Deductive Databases and Logic Programming. Morgan Kaufmann, Los Altos, CA, 1988, pp. 89–148.

[2] K.R. Apt and D. Pedreschi, *Reasoning about Termination of Pure Prolog Programs,* Information and Computation **106** (1993), 109–157.

[3] M. Bezem, *Characterizing Termination of Logic Programs with Level Mappings.* In: E.L. Lusk and R.A. Overbeek (Eds.), Proceedings of the North American Conference on Logic Programming, MIT Press, Cambridge, MA, 1989, pp. 69–80.

[4] H.A. Blair, F. Dushin, D.W. Jakel, A.J. Rivera and M. Sezgin, *Continuous Models of Computation for Logic Programs.* In: K.R. Apt, V.W. Marek, M. Truszczynski and D.S. Warren (Eds.), The Logic Programming Paradigm: A 25-Year Perspective. Springer, Berlin, 1999, pp. 231–255.

[5] H.A. Blair and J. Remmel, *Hybrid Automata: Convergence Spaces and Continuity.* Proceedings of the joint IIIS & IEEE meeting of the 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI2001) and the 7th International Conference on Information Systems Analysis and Synthesis (ISAS2001), Orlando, Florida, USA, July, 2001. International Institute of Informatics and Systemics, 2001.

[6] A. Batarek, *Topological Aspects of Logic Programming.* PhD thesis, Syracuse University, June 1989.

[7] A. Batarekh and V.S. Subrahmanian, *Topological Model Set Deformations in Logic Programming,* Fundamenta Informaticae **12** (3) (1989), 357–400.

[8] A. Batarekh and V.S. Subrahmanian, *The Query Topology in Logic Programming.* In: Proc. 1989 Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Compuer Science Vol. 349, Springer, Berlin, 1989, pp. 375–387.

[9] A. Bossi, M. Gabbrielli, G. Levi and M. Martelli, *The* **s**-*Semantics Approach: Theory and Applications,* Journal of Logic Programming **19 & 20** (1994), 149–198.

[10] L. Cavedon, *Acyclic Programs and the Completeness of SLDNF-Resolution,* Theoretical Computer Science **86** (1991), 81–92.

[11] K.L. Clark, *Negation as Failure.* In: H. Gallaire and J. Minker (Eds.): Logic and Data Bases. Plenum Press, New York (1978), pp. 293–322.

[12] A. Edalat and R. Heckmann, *A Computational Model for Metric Spaces,* Theoretical Computer Science **193** (1998), 53–73.

[13] F. Fages, *A New Fixpoint Semantics for General Logic Programs compared with the Well-founded and the Stable Model Semantics,* New Generation Computing **9** (1991), 425–443.

[14] F. Fages, *Consistency of Clark's Completion and Existence of Stable Models,* Journal of Methods of Logic in Computer Science **1** (1994), 51–60.

[15] M. Fitting, *A Kripke-Kleene-Semantics for General Logic Programs,* Journal of Logic Programming **2** (1985), 295–312.

[16] M. Fitting, *Metric Methods: Three Examples and a Theorem,* J. Logic Programming **21** (3) (1994), 113–127.

[17] A. Van Gelder, K.A. Ross and J.S. Schlipf, *The Well-Founded Semantics for General Logic Programs,* Journal of the ACM **38** (3) (1991), 620–650.

[18] G. Gelfond and V. Lifschitz, *The Stable Model Semantics for Logic Programming.* In: R.A. Kowalski, K.A. Bowen (Eds.), Logic Programming. Proceedings of the 5th International Conference and Symposium on Logic Programming, MIT Press, 1988, pp. 1070–1080.

[19] M.A. Gutiérrez Naranjo, J.A. Alonso Jiménez and J. Borrego Díaz, *A Topological Study of the Upward Refinement Operators in ILP.* In: J. Cussens and A. Frisch (Eds.), Inductive Logic Programming, 10th International Conference (ILP2000) Work-in-Progress Reports, London, 2000, pp. 120–137.

[20] P. Hitzler, *Generalized Metrics and Topology in Logic Programming Semantics.* PhD thesis, Department of Mathematics, National University of Ireland, University College Cork, 2001.

[21] P. Hitzler and A.K. Seda, *Characterizations of Classes of Programs by Three-Valued Operators.* In: M. Gelfond, N. Leone and G. Pfeifer (Eds.), Logic Programming and Non-Monotonic Reasoning, Proceedings of the 5th International Conference on Logic Programming and Non-Monotonic Reasoning (LPNMR'99), El Paso, Texas, USA, December 1999. Lecture Notes in Artificial Intelligence, Vol. 1730. Springer, Berlin (1999), 357–371.

[22] P. Hitzler and A.K. Seda, *The Fixed-Point Theorems of Priess-Crampe and Ribenboim in Logic Programming*. Proceedings of the International Conference and Workshop in Valuation Theory in Honour of Paulo Ribenboim, Saskatchewan, Canada, July 1999. Fields Institute Communications Series, American Mathematical Society, pp. 1–17, to appear.

[23] P. Hitzler and A.K. Seda, *Acceptable Programs Revisited*. Proceedings of the Workshop on Verification in Logic Programming, Sixteenth International Conference on Logic Programming (ICLP'99), Las Cruces, New Mexico, November, 1999. Electronic Notes in Theoretical Computer Science, Volume 30, No. 1, Elsevier, 1999, pp. 1–18.

[24] P. Hitzler and A.K. Seda, *Dislocated Topologies.* Proceedings of the Slovakian Conference in Applied Mathematics, Bratislava, 2000. Journal of Electrical Engineering, Vol. 51 No. 12/s, Slovak Academy of Sciences (2000), 3–7.

A more detailed version of this paper is available from the authors' web pages as a Technical Report with the same title, Department of Mathematics, University College Cork, 17 pages.

[25] P. Hitzler and A.K. Seda, *Unique Supported-Model Classes of Logic Programs,* Information **4** (3), International Information Institute, 2001, 295-302.

[26] P. Hitzler and A.K. Seda, *A New Fixed-point Theorem for Logic Programming Semantics.* Proceedings of the joint IIIS & IEEE meeting of the 4th World Multiconference on Systemics, Cybernetics and Informatics (SCI2000) and the 6th International Conference on Information Systems Analysis and Synthesis (ISAS2000), Orlando, Florida, USA, July 2000. International Institute of Informatics and Systemics: IIIS, Vol. VII, Computer Science and Engineering Part 1, 2000, pp. 418–423.

[27] P. Hitzler and A.K. Seda, *On the Coincidence of Semantics for Uniquely Determined Programs.* In: T. Hurley, M. Mac an Airchinnigh, M. Schellekens and A.K. Seda (Eds.), Proceedings of the First Irish Conference on the Mathematical Foundations of Computer Science and Information Technology, Cork, Ireland, July 2000. Electronic Notes in Theoretical Computer Science, Volume 40, 2001, Elsevier, pp. 1–17.

[28] P. Hitzler and A.K. Seda, *A Note on the Relationships between Logic Programs and Neural Networks.* In: P. Gibson and D. Sinclair (Eds.), Proceedings of the 4th Irish Workshop on Formal Methods (IWFM'00), Maynooth, July 2000. Electronic Workshops in Computing (eWiC), British Computer Society, pp. 1–9.

[29] S. Hölldobler, H. Störr and Y. Kalinke, *Approximating the Semantics of Logic Programs by Recurrent Neural Networks,* Applied Intelligence **11** (1999), 45–58.

[30] S. Hölldobler, *Challenge Problems for the Integration of Logic and Connectionist Systems.* Technical Report WV-99-03, Artificial Intelligence Institute, Department of Computer Science, Dresden University of Technology, 1999.q

[31] M.A. Khamsi, V. Kreinovich and D. Misane, *A New Method of Proving the Existence of Answer Sets for Disjunctive Logic Programs: A Metric Fixed-Point Theorem for Multivalued Mappings.* In: C. Baral and M. Gelfond (Eds.), Proceedings of the Workshop on Logic Programming with Incomplete Information, Vancouver, B.C., Canada, October 1993, pp. 58–73.

[32] J.W. Lloyd, *Foundations of Logic Programming.* Springer, Berlin, 1988.

[33] S.G. Matthews, *Metric Domains for Completeness.* PhD thesis (1985). Research Report 76, Department of Computer Science, University of Warwick, UK, April, 1986.

[34] S.G. Matthews, *Partial Metric Topology.* In: Proceedings of the 8th Summer Conference on Topology and its Applications, Annals of the New York Academy of Sciences, Vol. 728, 1992, pp. 183–197.

[35] S. Priess-Crampe and P. Ribenboim, *Ultrametric Spaces and Logic Programming,* J. Logic Programming **42** (2000), 59–70.

[36] H. Przymusinska and T.C. Przymusinski, *Weakly Stratified Logic Programs.* Fundamenta Informaticae **13** (1990), 51–65. K.R. Apt (Ed.), special issue on Logical Foundations of Artificial Intelligence.

[37] T.C. Przymusinski, *On the Declarative Semantics of Deductive Databases and Logic Programs.* In: J. Minker (Ed.), Foundations of Deductive Databases and Logic Programming. Morgan Kaufmann, Los Altos, CA 1988, pp. 193–216.

[38] R. Reiter, *A Logic for Default Reasoning,* Artificial Intelligence **13** (1980), 81–132.

[39] W.C. Rounds and G-Q. Zhang, *Clausal Logic and Logic Programming in Algebraic Domains.* To appear in Information and Computation.

[40] J.J.M.M. Rutten, *Elements of Generalized Ultrametric Domain Theory,* Theoretical Computer Science **170** (1996), 349–381.

[41] A.K. Seda, *Topology and the Semantics of Logic Programs,* Fundamenta Informaticae **24** (4) (1995), 359–386.

[42] A.K. Seda, *Quasi-metrics and the Semantics of Logic Programs,* Fundamenta Informaticae **29** (1) (1997), 97–117.

[43] A.K. Seda and P. Hitzler, *Topology and Iterates in Computational Logic.* Proceedings of the 12th Summer Conference on Topology and its Applications: Special Session on Topology in Computer Science, Ontario, August 1997. Topology Proceedings Vol. 22, Summer 1997, 427–469.

[44] A.K. Seda and P. Hitzler, *Strictly Level-Decreasing Logic Programs.* In: A. Butterfield and S. Flynn (Eds.), Proceedings of the Second Irish Workshop on Formal Methods (IWFM'98), Cork, 1998, Electronic Workshops in Computing, British Computer Society, 1999, pp. 1–18.

[45] M.B. Smyth, *Quasi-uniformities: Reconciling Domains with Metric Spaces.* In: M. Main, A. Melton, M. Mislove and D. Schmidt (Eds.), Mathematical Foundations of Programming Language Semantics, Lecture Notes in Computer Science Vol. 198, Springer, Berlin, 1987, pp. 236–253.

[46] M.B. Smyth, *Totally Bounded Spaces and Compact Ordered Spaces as Domains of Computation.* In: G.M. Reed, A.W. Roscoe and R.F. Wachter (Eds.), Topology and Category Theory in Computer Science. Oxford University Press, 1991, pp. 207–229.

[47] V. Stoltenberg-Hansen, I. Lindström and R. Griffor, *Mathematical Theory of Domains.* Cambridge University Press, 1994.

[48] S. Willard, *General Topology.* Addison-Wesley, Reading MA, 1970.

[49] G.-Q. Zhang, *Logic of Domains.* Progress in Theoretical Computer Science, Birkhäuser, Boston, 1991.

[50] G-Q. Zhang and W.C. Rounds, *Complexity of Power Default Reasoning.* In: Proceedings of the Twelfth Annual IEEE Symposium on Logic in Computer Science, LICS'97, Warsaw, Poland, July 1997. IEEE Computer Society Press pp. 328–339.

[51] G-Q. Zhang and W.C. Rounds, *Power defaults (preliminary report).* In: J. Dix, U. Furbach and A. Nerode (Eds.), Proceedings of the Fourth International Conference on Logic Programming and Non-Monotonic Reasoning (LPNMR'97), Dagstuhl, Germany, July 28-31, 1997. Lecture Notes in Computer Science, Vol. 1265, Springer-Verlag, 1997, pp. 152–169.

[52] G-Q. Zhang and W.C. Rounds, *Semantics of Logic Programs and Representation of the Smyth Powerdomain.* In: K. Keimel, G.-Q. Zhang et al (Eds.), Domains and Processes (ISDT'99), volume 1 of the series *Semantic Structures in Computation*, Kluwer Academic Publishers, pp. 151-179, 2001.

A Appendix: More on the Atomic Topology

We briefly list here some aspects of the atomic topology which the topologically minded reader may find interesting. We refer to [41, 20] for proofs and further details. In the following, P is a normal logic program. Also, we work over arbitrary preinterpretations, so that $\mathcal{B}_P$, the set of all ground instances of atoms in that preinterpretation[9], may be uncountable. The same applies to ground(P). Again, I_P denotes the set of all interpretations over the given preinterpretation. The single-step operator T_P is defined exactly as in the Herbrand case.

For every literal L, let $\mathcal{G}(L) = \{I \in I_P \mid I \models L\}$ and form the sets $\mathcal{G}^+ = \{\mathcal{G}(A) \mid A \in B_P\}$ and $\mathcal{G}^- = \{\mathcal{G}(\neg A) \mid A \in B_P\}$. Then $\mathcal{G}^+$ is a subbase of the Scott-topology on I_P, while $\mathcal{G}^+ \cup \mathcal{G}^-$ is a subbase of the atomic topology Q on I_P. We note that the basic open sets of Q are of the form $\mathcal{G}(A_1, \dots, A_k, \neg B_1, \dots, \neg B_m) := \mathcal{G}(A_1) \cap \dots \cap \mathcal{G}(A_k) \cap \mathcal{G}(\neg B_1) \cap \dots \cap \mathcal{G}(\neg B_m)$.

The atomic topology can be characterized in terms of convergence as follows.

A.1 Proposition A net (I_λ) converges in Q to $I \in I_P$ if and only if every element in I is eventually in I_λ and every element not in I is eventually not in I_λ, that is, for each $A \in I$ there exists λ_0 such that for all $\lambda \geq \lambda_0$ we have $A \in I_\lambda$ and for each $A \in B_P$ with $A \notin I$ there exists λ_1 such that for all $\lambda \geq \lambda_1$ we have $A \notin I_\lambda$.

The following result records some basic facts about Q.

A.2 Theorem The topology Q on I_P coincides with the product topology on $\mathbf{2}^{B_P}$, where $\mathbf{2} = \{0, 1\}$ is endowed with the discrete topology. Thus, (I_P, Q) is

[9] As usual, $\mathcal{B}_P$ contains all formal symbols $p(d_1, \dots, d_n)$ for which p is a predicate symbol from P and $d_1, \dots, d_n$ are elements of the domain of the preinterpretation.

a totally disconnected compact Hausdorff space. It is also second countable and metrizable if the domain of the chosen preinterpretation is countably infinite and in that case is homeomorphic to the Cantor set in the real line.

We finally present some results which underline the importance of the atomic topology as an alternative to the Scott topology when non-monotonicity of operators is present.

A.3 Theorem The following hold.

(i) If, for some $I \in I_P$, the sequence $(T_P^n(I))$ converges in Q to an interpretation M, then M is a model for P.

(ii) If the sequence $(T_P^n(I))$ does not converge in Q for any $I \in I_P$, then P has no supported models.

Let P be a normal logic program and let $I \in I_P$ be such that the sequence $(T_P^n(I))$ converges in Q to some $M \in I_P$. Then, by Theorem A.3, M is a model for P. If, furthermore, T_P is continuous in Q, or at least continuous at M in Q, then $M = \lim T_P^{n+1}(I) = \lim T_P(T_P^n(I)) = T_P(\lim T_P^n(I)) = T_P(M)$. So M is a supported model in this case.

The following result characterizes continuity in Q.

A.4 Theorem The single-step operator T_P is continuous in Q if and only if, for each $I \in I_P$ and for each $A \in B_P$ with $A \notin T_P(I)$, either there is no clause in P with head A or there is a finite set $S(I, A) = \{A_1, \ldots, A_k, B_1, \ldots, B_{k'}\}$ of elements of B_P with the following properties:

(i) $A_1, \ldots, A_k \in I$ and $B_1, \ldots, B_{k'} \notin I$.

(ii) Given any clause C with head A, at least one $\neg A_i$ or at least one B_j occurs in the body of C.

As a corollary, one obtains that programs without local variables[10] have continuous single-step operators, and also that the single-step operator is not in general continuous for arbitrary programs.

A.5 Theorem Let P be a normal logic program and let $I_0 \in I_P$ be such that the sequence (I_n), with $I_n = T_P^n(I_0)$, converges in Q to some $M \in I_P$. If, for every $A \in M$, no clause whose head matches A contains a local variable, then M is a supported model.

[10] A variable is *local* if it occurs in the body of a clause, but not in its head.

The following result is an obvious, but fundamental, generalization of Theorem A.3 prompted by the observation that transfinite iterations of the single-step operator are sometimes necessary[11] in order to achieve a fixed point.

A.6 Theorem Let P be a normal logic program and let $I \in I_P$. Define, for each limit ordinal α,

$$T_P^{\alpha}(I) = \left\{ A \in B_P \mid A \text{ is eventually in } \left(T_P^{\beta}(I)\right)_{\beta<\alpha} \right\}.$$

If, for some limit ordinal γ_0, the transfinite sequence $(T_P^{\gamma}(I))_{\gamma<\gamma_0}$ converges in Q, then the limit of this sequence is a model for P.

[11] For example, for locally hierarchical programs in general.