Advance Access publication on January 17, 2008 doi:10.1093/comjnl/bxm108

Generalized Distance Functions in the Theory of Computation

Anthony Karel Seda[1,*] and Pascal Hitzler[2]

[1]Department of Mathematics, University College Cork, Cork, Ireland
[2]Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany
*Corresponding author: a.seda@ucc.ie

We discuss a number of distance functions encountered in the theory of computation, including metrics, ultra-metrics, quasi-metrics, generalized ultra-metrics, partial metrics, d-ultra-metrics and generalized metrics. We consider their properties, associated fixed-point theorems and some general applications they have within the theory of computation. We consider in detail the applications of generalized distance functions in giving a uniform treatment of several important semantics for logic programs, including acceptable programs and natural generalizations of them, and also the supported model and the stable model in the context of locally stratified extended disjunctive logic programs and databases.

Keywords: generalized distance functions; ultra-metrics; topology; fixed-point theorems; denotational semantics; logic programming; supported model; stable model

Received 22 August 2007; revised 14 November 2007

1. INTRODUCTION

The concepts of (1) the 'distance between two objects' and (2) whether or not two objects are in some sense 'close' are fundamental both as mathematical ideas in themselves and in the many and varied applications of mathematics to other subjects. Therefore, it is not surprising that such ideas are important within a number of areas of information theory and the theory of computation.

These two concepts can be formalized as follows. First, at a completely general level, a (*generalized*) *distance function* d defined on a set X is simply a mapping $d : X \times X \to A$, where A is some suitable set of values (a *value set*), and the distance between x and y is taken to be the element $d(x, y)$ of A. Second, and again at a completely general level, closeness can be defined by assigning to each element x of a set X a family $\mathcal{U}_x$ of subsets U of X called *neighbourhoods* of x; then y can be thought of as *close to* x if y belongs to some neighbourhood U of x. Under suitable restrictions these notions are very familiar in mathematics leading on the one hand to metrics, ultra-metrics, pseudo-metrics and the like, and on the other hand to topologies. In turn, these concepts have many ramifications including: (i) *fixed points of functions* f defined on a set X, say, or in other words elements x of X with the property that $f(x) = x$ (x and $f(x)$ perhaps being thought of as zero distance apart), and (ii) limits of sequences and nets (or filters) for describing convergence.

The level of generality just considered is too high to be useful without some conditions being imposed on d or on A or on both d and A. Furthermore, in many ways the notions of 'distance between two objects' and 'closeness' are synonymous. Therefore, conditions on d and A on the one hand should correspond to conditions on the families $\mathcal{U}_x$ of neighbourhoods on the other. If one asks what distance functions d are generally appropriate in mathematical analysis, say, the answer is relatively simple: metrics (perhaps derived from norms on vector spaces) and families of seminorms. On the other hand, the question of appropriate limits to the generality of d and of A in the definition of a distance function in relation to computation is not so easy to answer. This is partly due to the diversity of situations encountered in computation, and we discuss this point in the next paragraph. One pointer in the direction of such appropriate limits is provided by Smyth in his chapter on topology in [1], in which he discusses observable properties. Here, one envisages a black box outputting a binary sequence in the presence of an observer, and it is shown that the class of properties which the observer can verify forms a topology, called the topology of observable properties. Furthermore, a number of important connections between topology and computer science are discussed in [1]. However, useful examples of structures more general than topologies have been encountered within the semantics of computation; see [2]. Nevertheless, it is convenient to take as a starting point in discussing distance functions and families of neighbourhoods, that level of generality which corresponds to conventional topologies (or equivalently, their associated neighbourhoods) and gives an equivalence between these latter notions and that of distance function. Therefore, our conceptual framework can be viewed as being that of continuity spaces and continuity functions [3], since these give precisely the equivalence just mentioned. However, we make no real use of continuity spaces and simply show that each distance function we consider in the paper, at least in the form in which we use it, actually is a continuity function.

There are many non-trivial applications of distance functions to computer science and information theory in general, some of them quite old and some more recent. Indeed, the following list is quite long, although by no means exhaustive: the use of ultra-metrics in the study of infinite trees by Arnold and Nivat in [4], and their use in non-determinism in [5] (see [1, Section 6.2] for further, related examples); the use of ultra-metrics in cognitive information (see [6–8]), in time series (see [9, 10]) and in bioinformatics (see [11, 12]); the use of metrics in studying processes and concurrency [13] (see also the articles in [14] by Barrett and Goldsmith and by de Bakker and Rutten); the many uses of the Hamming distance and other (pseudo-)metrics in information theory, and elsewhere in measuring the distance between logical formulae (for an interesting application to neural networks, see [15]); the use of distance functions in deriving fixed-point theorems and their appplications to the semantics of programs and their correctness, and the proof of program properties; attempts to measure the 'distance' between programs, and attempts to make quantitative statements about processing speed, speed of convergence and complexity of programs and algorithms by means of partial metrics and (weighted) quasi-metrics in quantitative domain theory; the use of quasi-metrics in abstract interpretation, and in access prediction in the context of replicated databases. In Section 7, we comment further on the more recent of these applications of distance functions, but the reader should also consult the companion papers in this volume for more detailed information on these and other applications of distance functions. In addition, there is the overall question of unifying the qualitative (order-theoretic) view of computation and the quantitative (distance-theoretic) view by means of suitable distance functions, and we consider this point in Section 2. Thus, within computing, there is a wide variety both of distance functions of various types and of their applications.

We will concentrate here on the use of distance functions within the semantics of computation and particularly within the semantics of logic programming, and the reason for this is as follows. In conventional programming language semantics, such as the denotational semantics of functional and imperative programs, fixed points of operators (and of functors) play an important role, and indeed are fundamental wherever recursion and self-reference are encountered. However, in that context the operators which arise are usually monotonic, indeed continuous. Therefore, the main fixed-point theorem in general use in classical semantics is the well-known Knaster–Tarski theorem based on order theory, which we state as follows: if T is defined and monotonic on a complete partial order X, then T has a least fixed point which is also the least pre-fixed point of T. In fact, if T is continuous, then the least fixed point of T is the supremum of the set of iterates $T^n(\bot)$, where $\bot$ denotes the bottom element of X; see [16]. This latter statement is sometimes referred to as Kleene's theorem or the fixed point-theorem, and we adopt this nomenclature here. Furthermore, the same sort of representation of the least fixed point of T can even be obtained for arbitrary monotonic T if one works transfinitely with ordinal powers; see [17]. On the other hand, the situation in the semantics of logic programs is rather different. Once one introduces negation, which is certainly desirable from the point of view of expressiveness and enhanced syntax, then certain of the important operators associated with logic programs are not monotonic and therefore not continuous (see Section 3), and in consequence neither the Knaster–Tarski theorem nor Kleene's theorem is applicable to them. Various ways have been proposed to overcome this problem. One such is to introduce syntactic conditions on programs, see [18, 19] for example; and to disallow those programs not meeting these conditions, in an attempt to recover continuity in the order-theoretic sense. Another is to consider different operators, and we discuss this later. The third main solution is to introduce techniques from topology and analysis to augment arguments based on order. Thus, one finds methods based on topology [20–26], on metrics [27–29], on quasi-metrics [30, 31], on ultra-metrics and on d-ultra-metrics, as we see later. Indeed, logic programming semantics is a very fertile area in respect of the use of various distance functions in its study.

Thus, the purpose of this paper is to discuss the role of distance functions and their applications in general within the theory of computation, with special emphasis on logic programming semantics, including the roles of the associated topologies and fixed-point theorems. An especially important consideration as we proceed is the provision of various conditions and restrictions that one can place on distance functions, and the corresponding effect these have on applications within the theory of computation. This includes, in particular, the important issue of the provision of fixed-point theorems or, in other words, the determination of conditions on d and A which guarantee that functions $f : X \to X$ have fixed points. Throughout, we will make considerable use of elementary ideas from order theory, and we refer the reader to [32] for background in this subject.[1]

[1]Ordered sets, in particular complete lattices, play a fundamental role in topics such as Formal Concept Analysis; see [33]. In turn, the interplay between topology and order, see for example [34], suggests a link between Formal Concept Analysis and the topic of this paper, and indeed some use of distance functions has already been made in Formal Concept Analysis; see [35]. We are grateful to one of the referees for drawing our attention to this link. At the same time, interrelations between logic programming, Formal Concept Analysis, and domain theory have been studied—albeit not from a metric perspective—in [36, 37].

It is convenient to divide the paper into two parts, Part I in which we consider, in their own right, many of the main distance functions encountered in the theory of computation, and Part II in which we consider some substantial applications of these to logic programming semantics. Thus, the structure of the paper is as follows. In Section 2 of Part I, we briefly summarize the result of Kopperman [3] that all topologies come from generalized distance functions via continuity spaces. As already noted, we view this as providing a uniform and, for our purposes, sufficiently general setting in which to discuss distance functions. Following this, we consider a number of specific distance functions, including: metrics, ultra-metrics, quasi-metrics, generalized ultra-metrics, partial metrics, d-ultra-metrics, and generalized metrics (in the sense of Khamsi, Kreinovich and Misane), together with their properties, associated fixed-point theorems and some general applications they have. In Part II, we discuss the applications of some of the results of Part I in deriving several of the important standard fixed-point semantics encountered in logic programming, as follows. In Section 4, we derive in detail the semantics of Φ^*-accessible programs, an important class containing the acceptable programs of [38]. In Section 5, we show in summary, giving references to the proofs, that every locally stratified program has a supported model, and that every locally hierarchical program has a unique supported model (its perfect model). In Section 6, we show, again in summary, that every locally stratified extended disjunctive logic program (or database) admits a stable model. A certain minimum amount of background and notation from logic programming is needed, and this we present in Section 3. It should be noted that the original derivation of the various semantics just listed was by completely different means. Therefore, what we illustrate here is the application of distance functions in obtaining a unified approach to the fixed-point theory of very general and significant classes of logic programs and databases. Finally, in Section 7, we summarize other, recent applications of various distance functions within the theory of computation, and in Section 8 we present our conclusions.

The main results and applications we discuss here involve ultra-metrics (and ultra-metric topology) or generalized ultra-metrics. Therefore, overall, the paper can be viewed as making a contribution to the theory of programming languages within the general theme of ultra-metric information theory.

PART I: GENERALIZED DISTANCE FUNCTIONS

2. DISTANCE FUNCTIONS

In this section, we discuss distance functions in considerable generality including, we believe, most of the important special cases of them arising in computer science. It is well known that the fixed points of operators determined by algorithms and programs are fundamental in studying their semantics, and hence we include also the main fixed-point theorems associated with the various distance functions we consider.

We begin by sketching the result of [3] that every topology arises by means of some generalized distance function, in the setting of continuity spaces. We refer also to [39] and related papers where the notion of continuity space has been developed further in a number of directions, and to [40] for further background.

2.1. The generality of distance functions

It will be convenient to start with the definition of a topology on a set X.

DEFINITION 2.1. *By a* topology $\mathcal{T}$ *on a set* X *we mean a collection of subsets of* X *containing the empty set* Ø *and* X *itself and closed under the formation of finite intersections and arbitrary unions of its members. Thus,* Ø, X, $O_1 \cap O_2$ *and* $\cup_{i\in\mathcal{I}} O_i$ *are elements of* $\mathcal{T}$ *whenever* $O_1, O_2 \in \mathcal{T}$ *and* $\{O_i \mid i \in \mathcal{I}\}$ *is any collection of elements of* $\mathcal{T}$. *The elements of* $\mathcal{T}$ *are called* open *sets. A subset* U *of* X *is called* a neighbourhood *of an element* x *of* X *if there is an open set* $O \in \mathcal{T}$ *such that* $x \in O \subseteq U$.

Of course, neighbourhoods, as defined here, satisfy certain properties one might consider to be characteristic of closeness: for example, if U is a neighbourhood of x, there is some neighbourhood V of x such that if $y \in V$, then there is a neighbourhood W of y satisfying $W \subseteq U$. Indeed, the concept of neighbourhood can be taken as fundamental and that of topology as derived from it, see [41] for details.

Take for a moment the familiar case of distance functions d which are metrics; see Definition 2.6 and the remark following it. Thus, the usual value set A of d in this case is the interval $[0, \infty)$. Given some real number $\varepsilon > 0$, one defines the (*open*) *ball* $N_\varepsilon(x)$ *of radius* ε *about a point* $x \in X$ by setting $N_\varepsilon(x) = \{y \in X \mid d(x, y) < \varepsilon\}$. A subset O of X is then declared to be *open* if, for each $x \in X$, there is some $\varepsilon > 0$ such that $N_\varepsilon(x) \subseteq O$. It is easy to see that the collection of such open sets O forms a topology on X. Notice that in defining 'open' sets O here, one can equivalently require $B_{\varepsilon'}(x) \subseteq O$ for suitable $\varepsilon' > 0$, where $B_\varepsilon(x) = \{y \in X \mid d(x, y) \leq \varepsilon\}$ denotes the (*closed*) *ball of radius* ε *about a point* $x \in X$.

However, it is not true that every topology on X arises thus via a metric d and, for example, this statement applies to the Scott topology on a directed complete partial order; see Definition 2.8 for the definition of the Scott topology. Nevertheless, as already noted, every topology can be generated by means of a suitable distance function. Indeed, following [3], we next consider briefly the details of one way of establishing this claim, beginning with several definitions.

DEFINITION 2.2. *A* value semigroup A *is an additive abelian semigroup with identity* 0 *and absorbing element* ∞,[2] *where* $\infty \neq 0$, *satisfying the following axioms.*

[2] An element satisfying $a + \infty = \infty + a = \infty$ for all $a \in A$.

(1) *For all* $a, b \in A$, *if* $a + x = b$ *and* $b + y = a$ *for some* $x, y \in A$, *then* $a = b$.
(*Note that, using this property, we can define a partial order* $\leq$ *on A by setting* $a \leq b$ *if and only if* $b = a + x$ *for some* $x \in A$*; we call* $\leq$ *the* partial order induced on A by the operation $+$.)
(2) *For each* $a \in A$, *there is a unique* b $(=a/2) \in A$ *such that* $b + b = a$.
(3) *For all* $a, b \in A$, *the infimum* $a \wedge b$ *of a and b exists in A relative to the partial order* $\leq$ *defined in* (1).
(4) *For all* $a, b, c \in A$, $(a \wedge b) + c = (a + c) \wedge (b + c)$.
Note that if $\{(A_i, +_i, 0_i, \infty_i) | i \in \mathcal{I}\}$ is a family of value semigroups, then so is their product $(A, +, 0, \infty)$, where $+, 0, \infty$ are defined coordinatewise.

DEFINITION 2.3. *A set P of* positives *in a value semigroup A is a subset P of A satisfying the following axioms.*

(1) *If* $r, s \in P$, *then* $r \wedge s \in P$.
(2) *If* $r \in P$ *and* $r \leq a$, *then* $a \in P$.
(3) *If* $r \in P$, *then* $r/2 \in P$.
(4) *If* $a \leq b + r$ *for all* $r \in P$, *then* $a \leq b$.

EXAMPLE 2.1. *The set* $\mathcal{R}$ *of extended real numbers* $[0, \infty]$ *together with addition forms a value semigroup, the set* $(0, \infty]$ *is a set of positives for this example, and the induced partial order* $\leq$ *is the usual one on* $\mathcal{R}$.

DEFINITION 2.4. ([3]). *A* continuity space *is a quadruple* $\mathcal{X} = (X, d, A, P)$, *where X is a non-empty set, A is a value semigroup, P is a set of positives in A, and* $d : X \times X \to A$ *is a function, called a* continuity function, *satisfying the following axioms.*

(d1) *For all* $x \in X$, $d(x, x) = 0$.
(d2) *For all* $x, y, z \in X$, $d(x, z) \leq d(x, y) + d(y, z)$.

Finally, we define the topology generated by a continuity space.

DEFINITION 2.5. *Suppose that* $\mathcal{X} = (X, d, A, P)$ *is a continuity space. Let* $x \in X$ *and let* $b \in P$. *Then* $B_b(x) = \{y \in X \mid d(x, y) \leq b\}$ *is called the* ball *of radius b about x. The* topology $\mathcal{T}(\mathcal{X})$ generated by $\mathcal{X}$ *consists of all those subsets O of X satisfying the property: if* $x \in O$, *then* $B_b(x) \subseteq O$ *for some* $b \in P$.

The main result concerning continuity spaces is the following theorem.

THEOREM 2.1. ([3]). *Given a continuity space* $\mathcal{X} = (X, d, A, P)$, *the collection* $\mathcal{T}(\mathcal{X})$ *of subsets of X is a topology on X. Conversely, given a topology* $\mathcal{T}$ *on a set X, there is a continuity space* $\mathcal{X} = (X, d, A, P)$ *with the property that* $\mathcal{T} = \mathcal{T}(\mathcal{X})$.

Given a topology $\mathcal{T}$ on X, it is worth noting that the continuity space $\mathcal{X} = (X, d, A, P)$ with the property that $\mathcal{T} = \mathcal{T}(\mathcal{X})$ used in the proof in [3] of Theorem 2.1. is obtained by taking A to be the product of $\mathcal{T}$ copies of $\mathcal{R}$, and P to be the product of $\mathcal{T}$ copies of $(0, \infty]$. The continuity function d is defined coordinatewise by $d(x, y)(S) = d_S(x, y)$ for each $S \in \mathcal{T}$, where $d_S(x, y) = 0$ if ($x \in S$ implies $y \in S$), $d_S(x, y) = q$ otherwise, where q is an element of $(0, \infty]$ fixed once and for all.

2.2. Important cases of distance functions and corresponding fixed-point theorems

The results of the previous subsection are satisfactory in indicating the generality of distance functions, and in providing a framework within which to discuss them. However, it is usual to impose various conditions on the distance functions employed in practice, and we consider some of these next. In addition, once suitable conditions are imposed on distance functions, one can expect to be able to establish fixed-point theorems in their presence, and we present certain of these also. In fact, multivalued functions (see Section 2.2.2 for the definition) arise in a number of places of importance in our discussion, so some of the fixed-point theorems we discuss are given for multivalued mappings; in each case, they specialize to meaningful statements for single-valued functions also.

2.2.1. Metrics, ultra-metrics and quasi-metrics

In effect, the most familiar examples of distance functions occur in the setting obtained in Example 1 by taking the value semigroup A to be $\mathcal{R}$. Of course, the conditions (d1) and (d2) of Definition 2.4 are then perfectly meaningful.

DEFINITION 2.6. *Let* $d : X \times X \to [0, \infty]$. *Consider the following conditions on d, where x, y, z are arbitrary elements of X.*

(d3) $d(x, y) = d(y, x)$.
(d4) $d(x, y) = 0$ *implies* $x = y$.
(d5) $d(x, z) \leq \max\{d(x, y), d(y, z)\}$.
We call d: a metric *if it satisfies* (d1)–(d4); *an* ultra-metric *if it satisfies* (d1), (d3), (d4), *and* (d5); *a* pseudo-metric *if it satisfies* (d1)–(d3); *a* quasi-metric *if it satisfies* (d1), (d2) *and the following axiom*:

(d6) *if* $d(x, y) = d(y, x) = 0$, *then* $x = y$;
and an ultra-quasi-metric *if it satisfies* (d1), (d5), (d6). *We denote any of these structures by* (X, d), *where d is any one of the distance functions just defined, and the context will determine the exact nature of d.*

In Definition 2.6, it is convenient to take the codomain of d to be $[0, \infty]$ rather than the more usual $[0, \infty)$. Notice also that (d5) implies (d2), and hence an ultra-metric is a metric and, furthermore, all the notions just defined are continuity functions. Moreover, given a pseudo-metric d, there is a standard procedure for passing to a metric defined on the equivalence classes of the relation $\sim$ defined by $x \sim y$ if and only if $d(x, y) = 0$, and it often, although not always, suffices to work with this derived metric instead of with the pseudo-metric. Since any metric is a quasi-metric, the main notion emerging in this subsection for our purposes is that of quasi-metric, and there are good reasons for developing this notion further, as follows. First, there are many non-trivial applications of (ultra-)metrics in computing as already mentioned, and we will consider more later in this paper; in a general sense, quasi-metrics subsume these applications of (ultra-)metrics, of course. Second, two of the main spaces used in the semantics of programming languages are (i) metric spaces (and the Banach contraction mapping theorem, Theorem 2.3), see [13] for example, and (ii) Scott domains (and the fixed-point theorem) especially; see [42] and the many other papers of Scott on this latter subject. Third, as indicated in the Introduction, there has been a lot of interest in reconciling these two spaces for denotational semantics, and quasi-metrics have proved to be important in this respect; see [43, 44, 45] for example, see also [46] and [47] for a different viewpoint.

For a given quasi-metric d on X, there is an associated metric $d^\star$ defined on X by $d^\star(x, y) = \max\{d(x, y), d(y, x)\}$. One says that (X, d) is *totally bounded* if the metric space $(X, d^\star)$ is totally bounded; that is, given any $\varepsilon > 0$, there is a finite subset E of X with the property that for each y in X there is an x in E satisfying $d^\star(x, y) \leq \varepsilon$.

There are two interesting examples of quasi-metrics related to computer science discussed in [45] as follows; we will return to them again later.

EXAMPLE 2.2. ([45]). *Let* $(D, \sqsubseteq)$ *be an arbitrary partially ordered set and define* d *on* $D \times D$ *by*

$$d(x, y) = \begin{cases} 0 & \text{if } x \sqsubseteq y, \\ 1 & \text{otherwise.} \end{cases}$$

Then d is an ultra-quasi-metric, called the *discrete quasi-metric*, and is totally bounded if and only if D is finite.

Before presenting the second example, Example 2.4, it is necessary to include the definition of a domain in the form in which we will use it later in Part II.

DEFINITION 2.7. *A partially ordered set* $(D, \sqsubseteq)$ *is called a* Scott-Ershov domain *or simply a* domain *with set* D_C *of compact elements* (*see* [16]), *if the following conditions hold.*

(i) $(D, \sqsubseteq)$ *is a directed complete partial order* (dcpo), *that is, D has a bottom element* $\perp$, *and the supremum* sup *A exists for all directed subsets A of D.*
(ii) *The elements* $a \in D_C$ *are characterized as follows: whenever A is directed and* $a \sqsubseteq$ sup *A, then* $a \sqsubseteq x$ *for some* $x \in A$.
(iii) *For each* $x \in D$, *the set* $\text{approx}(x) = \{a \in D_C \mid a \sqsubseteq x\}$ *is directed and* $x = \sup\ \text{approx}(x)$ (*this property is called* algebraicity of *D*).
(iv) *If the subset A of D is consistent* (*there exists* $x \in D$ *such that* $a \sqsubseteq x$ *for all* $a \in A$), *then* sup *A exists in D* (*this property is called* consistent completeness of *D*).

The conditions in this definition ensure the existence and construction of fixed points of continuous functions and the existence of function spaces. Moreover, the compact elements provide an abstract notion of computability. As is well known, domains are an important means of providing structures for modelling computation and in providing spaces to support the denotational semantics approach to understanding programming languages; see [16].

EXAMPLE 2.3.

(i) *The set of all partial functions from* $\mathbb{N}^n$ *into* $\mathbb{N}$ *ordered by graph inclusion is a domain whose compact elements are the finite functions.*
(ii) *The set* $(I_P, \subseteq)$ *of interpretations for a logic program* P, *see Section 3, is a domain whose compact elements are the finite subsets of* B_P.[3]
(iii) *The set* $(I_{P,3}, \subseteq)$ *of three-valued interpretations, see Section 3, is a domain whose compact elements are the pairs* (C_1, C_2) *of disjoint finite subsets of* B_P.

Associated with a dcpo is its Scott topology, and we pause to give next the definition of the open sets in this topology.

DEFINITION 2.8. *A subset O of a* dcpo $(D, \sqsubseteq)$ *is called* Scott open *if it satisfies the following two conditions*: (i) *O is* upwards closed, *that is, whenever* $x \in O$ *and* $x \sqsubseteq y$, *we have* $y \in O$ *and* (ii) *whenever* $A \subseteq D$ *is directed and* $\sup A \in O$, *then* $A \cap O \neq \emptyset$.

EXAMPLE 2.4. ([45]). *Let* $(D, \sqsubseteq)$ *be any Scott domain and let* $r : D_C \to \mathbb{N}$ *be a map (a rank function) such that* $r^{-1}(n)$ *is a finite set for each* $n \in \mathbb{N}$. *Define* d *on* $D \times D$ *by*

$$\begin{aligned} d(x, y) = \ & \inf\{2^{-n} | e \leq x \text{ implies } e \leq y \\ & \text{for all } e \in D_C \text{ with } r(e) \leq n\}. \end{aligned}$$

Then d is an ultra-quasi-metric which induces the Scott topology of D and (D, d) is totally bounded.

In fact, it is usually the Scott topology which is employed to study domains. However, as we will see in Section 2.2.2, domains can be endowed with the structure of a spherically complete generalized ultra-metric space. Given that there are

[3]Given a logic program P, the *Herbrand base* B_P for P (*or more correctly for the underlying first-order language* $\mathcal{L}$ of P) is the set of all ground (or variable-free) atoms which can be formed by using predicate symbols from $\mathcal{L}$ with ground terms from $\mathcal{L}$ as arguments. Thus, for example, if p, f, a, and b are respectively a binary or two-place predicate symbol, a unary or one-place function symbol, and constant symbols, all in $\mathcal{L}$, then the corresponding elements of B_P are those (infinitely many) atoms of which the following are typical: $p(a, a)$, $p(a, b)$, $p(b, a)$, $p(b, b)$, $p(f(a), a)$, $p(a, f(a))$, $p(f(a), b)$,..., $p(f(f(a)), a)$, $p(f(f(a)), b)$, $p(a, f(f(a)))$..., $p(f(a), f(f(a)))$.... The reader should consult Section 3 for further details of these matters.

many ultra-metrics which are useful in theoretical computer science, as mentioned in the Introduction and in Section 7 (see also the results we consider in Section 2.2.2), it seems likely that generalized ultra-metric spaces, as well as quasi-metric spaces, may well be a useful complement to the Scott topology in studying domains.

Turning now to fixed points, we note that many fixed-point theorems in various settings are established by iterating on some suitable element and that the resulting sequence is required to converge in some sense. If this approach is to work, some notion of completeness is required. In the case of metrics, it is the familiar and elementary notion of completeness which is appropriate, namely, convergence of each Cauchy sequence. However, in the case of quasi-metrics the situation is a bit more complicated due to the non-symmetry of the distance function involved, and we consider this issue next; the resulting notions collapse to the familiar ones if the quasi-metric involved is actually a metric or ultra-metric.

Definition 2.9. *A sequence* (x_n) *in the quasi-metric space* (X, d) *is said to be:*

(1) *forward Cauchy if, for each* $\varepsilon > 0$, *there is a natural number k such that* $d(x_l, x_m) \leq \varepsilon$ *whenever* $k \leq l \leq m$;
(2) *bi-Cauchy if, for each* $\varepsilon > 0$, *there is a natural number k such that* $d(x_l, x_m) \leq \varepsilon$ *whenever* $k \leq l, m$.

There is also a notion of *backward Cauchy* sequence which is obtained by replacing $d(x_l, x_m)$ by $d(x_m, x_l)$ in the first part of this definition, though we have no need of it here. Indeed, in [45] the point is made that the computationally most significant of these concepts is that of forward Cauchy, and that all three are equivalent in the presence of total boundedness [45, Theorem 10].

In the general context of a quasi-metric space (X, d), the appropriate notion of 'limit' of a forward Cauchy sequence (x_n) seems to be as given in [48] and [45, Definition 11] and is as follows; it is important in the developments made in [43, 45, 48] and also in what we wish to discuss here; see especially Theorem 2.2.

Definition 2.10. *Let* (x_n) *be a forward Cauchy sequence in a quasi-metric space* (X, d). *A point* $x \in X$ *is a* limit *of* (x_n), *which we write as* $x = \lim_{n\to\infty} x_n$ *or simply* $x = \lim x_n$ *if, for every* $y \in X$, *we have* $d(x, y) = \lim_{n\to\infty} d(x_n, y)$. *The space X is said to be* complete *if every forward Cauchy sequence in X has a limit*.

Note that this definition can be made for arbitrary sequences (x_n). However, for forward Cauchy sequences (x_n) it is the case that the sequence $d(x_n, y)$ is itself Cauchy in the real line; see [48]. Thus, $\lim_{n\to\infty} d(x_n, y)$ exists in the extended real line relative to the usual metric, which is understood in Definition 2.10, and it follows that this definition is always meaningful for forward Cauchy sequences. Moreover, limits, in this sense, of forward Cauchy sequences are always unique when they exist. In [43] such limits are called *metric limits*. It should be further noted that this definition requires no underlying topology for its formulation and indeed Smyth in [1, 44, 45] and Bonsangue *et al.* in [43] have quite extensively examined the interplay between such limits and topological limits.

We next consider some concepts applying to mappings between quasi-metric spaces. Amongst these is that of continuity as defined by Rutten in [48] (see also [43]) which, as with limits, does not involve any topology in its formulation; in [43] it is referred to as *metric continuity*.

Definition 2.11. *Let* (X, d) *be a quasi-metric space and suppose* $f : X \to X$ *is a mapping. We say that:*

(1) *f is* non-expansive *if, for all* $x, y \in X$, *we have* $d(f(x), f(y)) \leq d(x, y)$,
(2) *f is* contractive *or is a* contraction *if there exists a positive number* $c < 1$ *such that, for all* $x, y \in X$, *we have* $d(f(x), f(y)) \leq c\, d(x, y)$,
(3) *f is* continuous *if, for all forward Cauchy sequences* (x_n) *and x in X, whenever* $\lim x_n = x$, *we have* $\lim f(x_n) = f(x)$.

We are now ready to give the statement of Rutten's theorem [48, Theorem 3.7]. First note that the proof given by Rutten does not make essential use of the ulta-metric condition (d5) and easily extends to quasi-metrics; see also [49, Page 6 and Theorem 6.3]. Therefore, we state the result for quasi-metrics rather than for ultra-quasi-metrics. Note also that if (X, d) is a quasi-metric space, then there is an associated order $\leq_X$ induced on X by $x \leq_X y$ if and only if $d(x, y) = 0$.

Theorem 2.2. *Let* (X, d) *be a complete quasi-metric space and suppose* $f : X \to X$ *is non-expansive*.

(1) *If f is continuous and there is an x in X with the property that* $d(x, f(x)) = 0$ (*that is,* $x \leq_X f(x)$), *then f has a fixed point which is the least fixed point above x in the order* $\leq_X$.
(2) *If f is continuous and contractive, then f has a unique fixed point*.

Part 1 of this theorem implies Kleene's theorem. Furthermore, the terms 'completeness' and 'contraction' used in relation to quasi-metrics have their usual meaning when specialized to metric spaces. Therefore, Part 2 of Theorem 2.2 contains, as a special case, Banach's well-known contraction mapping theorem which we state next for completeness.

Theorem 2.3. (Banach). *Suppose* $f : X \to X$ *is a contraction on a complete metric space X. 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* $x \in X$.

Notice that neither non-expansiveness nor contractivity of f imply continuity of f in general in the sense that continuity is employed in Theorem 2.2. Various implications between these and other concepts are examined in [43, 48].

Because of the comment immediately following its statement, Theorem 2.2 has turned out to be important in reconciling the metric and order-theoretic approaches to conventional programming language semantics. Furthermore, both Examples 2.2 and 2.4 have been considered in [31] in the context of logic programming semantics by means of Theorem 2.2. In particular, Example 2.2 was used in [31] to derive the basic fixed-point properties of the single-step operator T_P for definite logic programs P; see Theorem 3.1. Indeed, it turns out that T_P is always non-expansive and continuous relative to the quasi-metric of Example 2.2, and hence Theorem 2.2 is applicable and yields the fixed-point properties of T_P. Example 2.4 was used to define natural quasi-metrics arising from logic programs, which can be used in company with Theorem 2.2 in analysing the programs in question. However, we will postpone further discussion of the applications of the results in Part I to logic programming until we reach Part II.

2.2.2. *Generalized ultra-metric spaces*

The next concept we introduce is that of a generalized ultra-metric, following [50, 51]. Here, the distance function d takes values in a partially ordered set Γ with least element, and axioms (d1), (d3), (d4) and a suitably modified version of (d5) hold. Specifically, we make the following definition.

DEFINITION 2.12. *Let X be a set, and let Γ be a partially ordered set with least element* 0. *The pair (X, d) is called a* generalized ultra-metric space *or* gum *if $d : X \times X \to \Gamma$ is a function satisfying the following axioms for all $x, y, z \in X$ and $\gamma \in \Gamma$.*

(gum1) $d(x, x) = 0$.
(gum2) $d(x, y) = 0$ *implies* $x = y$.
(gum3) $d(x, y) = d(y, x)$.
(gum4) *If* $d(x, y) \leq \gamma$ *and* $d(y, z) \leq \gamma$, *then* $d(x, z) \leq \gamma$.

For $0 \neq \gamma \in \Gamma$ and $x \in X$, the set $B_\gamma(x) = \{y \in X \mid d(x, y) \leq \gamma\}$ is called a γ-ball *or just a* ball *in X with* centre *x and* radius γ.

Notice that at the level of generality of the previous definition, the function d this time is not a continuity function; that is, Γ need not be a value semigroup. However, in the applications we will actually consider, d will indeed be a continuity function.

Once again, a suitable form of completeness is needed, this time for generalized ultra-metrics, and this is provided by the notion of 'spherical completeness', as follows. A generalized ultra-metric space X is called *spherically complete* if $\cap\mathcal{C} \neq \emptyset$ for any chain $\mathcal{C}$ of balls in X, where the term 'chain of balls' means, of course, a set of balls which is totally ordered by inclusion. Note that for ultra-metric spaces, spherical completeness implies completeness, but not conversely, see [23, Proposition 10].

As mentioned earlier, we will be concerned at certain places with fixed points of multivalued mappings, that is, with mappings $f : X \to 2^X$, where 2^X denotes the power set of the set X. A *fixed point* of such a mapping f is a point $x \in X$ with the property that $x \in f(x)$. A multivalued mapping f is called *non-empty* if, for all $x \in X$, $f(x) \neq \emptyset$.

Whilst the standard notion of contraction involving a numerical constant $c < 1$ (see Definition 2.11) is not available in the context of generalized ultra-metric spaces, appropriate and useful contractivity notions for mappings defined on such spaces can be given as follows.

DEFINITION 2.13. *A mapping $f : X \to X$ on a generalized ultra-metric space X is called:*

(i) contracting (on X) *if, for all $x, y \in X$, we have $d(f(x), f(y)) \leq d(x, y)$,*
(ii) strictly contracting (on X) *if, for all $x, y \in X$ with $x \neq y$, we have $d(f(x), f(y)) < d(x, y)$,*
(iii) strictly contracting on orbits *if, for all $x \in X$ with $f(x) \neq x$, we have $d(f^2(x), f(x)) < d(f(x), x)$.*

One then has the following theorem, due to Priess-Crampe and Ribenboim [52].

THEOREM 2.4. *Let (X, d, Γ) be a spherically complete generalized ultra-metric space and let $f : X \to X$ be contracting on X and strictly contracting on orbits. Then f has a fixed point. If f is strictly contracting on X, then the fixed point is unique.*

For multivalued functions, the previous definition immediately generalizes as follows.

DEFINITION 2.14. *A multivalued mapping $f : X \to 2^X$ on a generalized ultra-metric space X is called:*

(i) contracting (on X) *if, for all $x, y \in X$ and for every $a \in f(x)$, there exists an element $b \in f(y)$ such that $d(a, b) \leq d(x, y)$,*
(ii) strictly contracting (on X) *if, for all $x, y \in X$ with $x \neq y$ and for every $a \in f(x)$, there exists a $b \in f(y)$ such that $d(a, b) < d(x, y)$,*
(iii) strictly contracting on orbits *if, for all $x \in X$ and for every $a \in f(x)$ with $a \neq x$, there exists a $b \in f(a)$ such that $d(b, a) < d(a, x)$.*

For a multivalued mapping $f : X \to 2^X$, let $\Pi_x = \{d(x, y) | y \in f(x)\}$, and for a subset $\Delta \subseteq \Gamma$ denote by $\min \Delta$ the set of all minimal elements of Δ.

The main theorem here is as follows.

THEOREM 2.5 (PRIESS-CRAMPE AND RIBENBOIM [50, (3.1)]). *Let (X, d) be a spherically complete generalized ultra-metric space. Let $f : X \to 2^X$ be a non-empty contraction which is strictly contracting on orbits, and assume that for every $x \in X$ the set $\min \Pi_x$ is finite and that every element of Π_x has a lower bound in $\min \Pi_x$. Then f has a fixed point.*

This result has several corollaries, both for multivalued mappings and for single-valued mappings, and we state next those that we need in the sequel. Note that Theorem 2.7 is a slight extension of Theorem 2.4.

THEOREM 2.6 (PRIESS-CRAMPE AND RIBENBOIM [50 (3.4)]). *Let (X, d) be spherically complete, and let Γ be narrow, that is, such that every trivially ordered subset of Γ is finite. Let $f : X \to 2^X$ be non-empty, strictly contracting on orbits and such that $f(x)$ is spherically complete for every $x \in X$. Then f has a fixed point*.

THEOREM 2.7 (PRIESS-CRAMPE AND RIBENBOIM [50, 53]). *Let (X, d) be a generalized ultra-metric space which is spherically complete, and let $f : X \to X$ be contracting on X. Then either f has a fixed point or there exists a ball $B_\pi(z)$ such that $d(y, f(y)) = \pi$ for all $y \in B_\pi(z)$. If, in addition, f is strictly contracting on orbits, then f has a fixed point. Finally, this fixed point is unique if f is strictly contracting on X.*

As already noted, the set Γ used here in general need not be a value semigroup. However, for the applications we have in mind our choice of Γ will be a value semigroup, and we consider this point next.

Let $\gamma > 1$ denote an arbitrary countable ordinal, and denote by Γ_γ the set $\{2^{-\alpha} | \alpha \leq \gamma\}$ of symbols $2^{-\alpha}$. Then Γ_γ is totally ordered by $2^{-\alpha} < 2^{-\beta}$ if and only if $\beta < \alpha$, and indeed $2^{-\gamma}$ is the bottom element of Γ_γ. (Notice that Γ_γ is really nothing other than $\gamma + 1$ endowed with the dual of the usual ordering, but it is convenient to use the symbols $2^{-\alpha}$ rather than the symbols α to denote typical elements, as will be seen later. Notice also that we regard an ordinal γ as the set of all ordinals n such that $n \in \gamma$, that is, the set of ordinals n such that $n < \gamma$.) We define the binary operation $+$ on Γ_γ by

$$2^{-\alpha} + 2^{-\beta} = \max\{2^{-\alpha}, 2^{-\beta}\},$$

and take $2^{-\gamma}$ as the identity and 2^{-0} as the absorbing element, noting that $2^{-\gamma} \neq 2^{-0}$ by our mild assumption that $\gamma > 1$, where 0 denotes the finite limit ordinal zero. Notice that we will sometimes also use 0 to denote $2^{-\gamma}$, where this does not cause confusion. Then Γ_γ is a value semigroup in which $a/2 = a$, where $a = 2^{-\alpha}$ denotes a typical element of Γ_γ, and moreover the partial order induced on Γ_γ by $+$ coincides with that already defined. Furthermore, the set $\{2^{-\alpha} | \alpha < \gamma\}$ is a set of positives in Γ_γ.

Using this construction we can turn a domain $(D, \sqsubseteq)$ into a generalized ultra-metric space essentially using the construction of Example 2.4, as follows.

DEFINITION 2.15. *Let $r : D_C \to \gamma$ be a function, again called a* rank function, *form the set Γ_γ, and suppose that r satisfies the condition*[4] *that for all $\beta < \gamma$ there exist $\beta' \geq \beta$ and $c \in D_C$ such that $r(c) = \beta'$. Define $d_r : D \times D \to \Gamma_\gamma$ by*

$$d_r(x, y) = \inf\{2^{-\alpha} | c \sqsubseteq x \text{ if and only if } c \sqsubseteq y \text{ for all } c \in D_C \text{ with } r(c) < \alpha\}.$$

[4] Mild conditions such as this prevent pathology arising by excluding the possibility that r is constantly zero; see also Example 2.4.

The following result was established in [54, Theorem 4.7].

THEOREM 2.8. *The space (D, d_r) is a spherically complete generalized ultra-metric*.

Indeed, (D, d_r) is called the generalized ultra-metric space *induced by r*. The intuition behind d_r is that two elements x and y of the domain D are 'close' if they dominate the same compact elements up to a certain rank (and hence *agree* in this sense up to this rank); the higher the rank giving agreement, the closer are x and y.

2.2.3. *Partial metrics and* d-*metrics*

It is perhaps surprising that distance functions which fail to satisfy axiom (d1) should prove to be of any interest. Nevertheless, there are several instances in computer science where distance functions satisfying $d(x, x) \neq 0$ arise, and we will examine two of them here. These discussions suggest that distance functions satisfying the axiom $d(x, x) \neq 0$ may have other interesting applications within computer science.

The first case we consider is that of the (weak) partial metrics defined next. These were introduced by Matthews in [55, 56] in connection with the semantics of data flow networks as studied by Kahn in [57].

DEFINITION 2.16. *Let X be a set, and let $d : X \times X \to [0, \infty]$ be a function. We call d a* partial metric *on X if it satisfies the following axioms, where x, y, z are arbitrary elements of X.*

(p1) $x = y$ *if and only if* $d(x, x) = d(x, y) = d(y, y)$.
(p2) $d(x, x) \leq d(x, y)$.
(p3) $d(x, y) = d(y, x)$.
(p4) $d(x, z) \leq d(x, y) + d(y, z) - d(y, y)$.

A weak partial metric *is a function d satisfying conditions* (p1), (p3) *and* (p4), *but not necessarily the condition* (p2) *of* small self-distances.

Partial metrics and weak partial metrics were also studied in [47, 58, 59]; in fact, in [59] partial metrics are allowed to take negative distances. A (weak) partial metric d need not satisfy axiom (d1), and so $d(x, x)$ need not be zero. Indeed, the value of $d(x, x)$ has been called the size of x by Matthews [56], and used to express the extent to which x is partially defined: x is totally defined if $d(x, x) = 0$. Thus, a (weak) partial metric is not a continuity function in the sense employed in Definition 2.4. Nevertheless, the set of balls it determines yields a topology, and thus (weak) partial metrics fall into our general framework in which distance functions correspond to topologies. Furthermore, strong relationships between the topologies arising from partial metrics and the topologies usually discussed in domain theory can be established; see for example [60, 56, 47].

Another example of the occurrence of distance functions failing to satisfy axiom (d1) is provided by *d-metrics* which we consider next. These were studied in [55], where they are called *metric domains*, and also in [23], where they are used in the context of logic programming semantics and will be discussed further here in that same context in Part II.

DEFINITION 2.17. *Let* $d : X \times X \to [0, \infty]$ *be a function. We call d a* d-metric *on X if it satisfies axioms* (d2)–(d4), *and call d a* d-ultra-metric *if it satisfies axioms (d3)–(d5).*

It is clear that any (weak) partial metric is a d-metric. Furthermore, it is routine to extend the usual notions of *limit of a sequence* (called *d-limits*), *Cauchy sequence* and *completeness* to d-metric spaces. Once that is done, one then obtains the following generalization of the Banach contraction mapping theorem to d-metric spaces, due to Matthews [55]; see also [23].

THEOREM 2.9. *Let* (X, d) *be a complete d-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* $x \in X$.

Again, d-metrics are not continuity functions, since in general they fail to satisfy axiom (d1). Furthermore, whilst it is true, as already noted, that the set of balls determined by a (weak) partial metric yields a topology in the conventional sense, this is not the case for d-metrics. However, given a d-metric d, one can associate with d a metric d' defined by setting $d'(x, y) = d(x, y)$ for $x \neq y$ and setting $d'(x, x) = 0$ for all $x \in X$. Then d' is complete if and only if d is complete, and a function f which is a contraction relative to d is a contraction relative to d'; see [23, Propositions 26 and 27]. Indeed, the notions of *d-topological space* and *d-neighbourhood system of a point x in a d-metric space* have been examined in [61] and shown to have very similar properties to conventional, topologies, respectively neighbourhood systems; for example, the property of conventional neighbourhoods quoted immediately following the definition of a topology, Definition 2.1, is essentially valid. In effect, therefore, d-metrics also fall within our general framework in which distance functions correspond to systems of neighbourhoods with natural properties. Further, purely topological results of this nature have been established in [62] under the name of relational topology, a concept which includes d-topological spaces.

One can also extend the definition of a generalized ultra-metric to obtain the definition of a *d-generalized ultra-metric*, or simply a *d-gum*, by dropping the axiom (gum1), but retaining axioms (gum2)–(gum4) in Definition 2.12. The concepts defined for generalized ultra-metric spaces then easily extend to d-gums, noting that γ-balls may be empty in the case of d-gums and hence in defining spherical completeness one needs to stipulate that the chain $\mathcal{C}$ consists of non-empty balls. Furthermore, the definitions made for mappings between gums also extend to d-gums.

The following lemma, proved in [23], is well known for ordinary ultra-metric spaces; see [52]. We include its short proof for the sake of completeness.

LEMMA 2.1. *Let* (X, d, Γ) *be a d-gum. 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 $b \in B_\alpha(x) \cap B_\beta(y)$. Then $d(a, x) \leq \alpha$ and $d(b, x) \leq \alpha$, hence $d(a, b) \leq \alpha \leq \beta$. Since $d(b, y) \leq \beta$, we have $d(a, y) \leq \beta$, and hence $a \in B_\beta(y)$, which proves the first statement. The second statement follows from (1) by symmetry. For the third statement, we note that $d(x, y) \leq d(x, y) = d(y, x)$, and it follows from (gum4) that $d(x, x) \leq d(x, y)$ for all $x, y \in X$. Hence, $x \in B_{d(x,\, x)}(x) \subseteq B_{d(x,\, y)}(x)$ and also $x \in B_{d(x,\, y)}(y)$, and therefore $B_{d(x,\, y)}(x) \cap B_{d(x,\, y)}(y) \neq \emptyset$. Statement (3) now follows from (2) on taking α to be equal to $d(x, y)$ in (2). □

The following theorem unites Theorem 2.9 of Matthews [55] and Theorem 2.4 of Priess-Crampe and Ribenboim [52]. The proof of the latter theorem given in [52] in fact carries over directly to our more general setting of d-gums, and we include it to illustrate the methods used.

THEOREM 2.10. *Let* (X, d, Γ) *be a spherically complete d-gum and let* $f : X \to X$ *be contracting on X and strictly contracting on orbits. Then f has a fixed point. If f is strictly contracting on X, then the fixed point is unique.*

Proof. Assume that f has no fixed point. Then for all $x \in X$, $d(x, f(x)) \neq 0$. We define the set $\mathcal{B}$ by $\mathcal{B} = \{B_{d(x,\, f(x))}(x) | x \in X\}$, and note that each ball in this set is non-empty. By Lemma 2.1(3), we know that $B_{d(x,\, f(x))}(x) = B_{d(x,\, f(x))}(f(x))$. Now let $\mathcal{C}$ be a maximal chain in $\mathcal{B}$. Since X is spherically complete, there exists a $z \in \cap\, \mathcal{C}$. We show that $B_{d(z,\, f(z))}(z) \subseteq B_{d(x,\, f(x))}(x)$ for all $B_{d(x,\, f(x))}(x) \in C$ and hence, by maximality, that $B_{d(z,\, f(z))}(z)$ is the smallest ball in the chain. Let $B_{d(x,\, f(x))}(x) \in \mathcal{C}$. Since $z \in B_{d(x,\, f(x))}(x)$, and noting our earlier observation that $B_{d(x,\, f(x))}(x) = B_{d(x,\, f(x))}(f(x))$ for all x, we obtain $d(z, x) \leq d(x, f(x))$ and $d(z, f(x)) \leq d(x, f(x))$. Since f is contracting, we get $d(f(z), f(x)) \leq d(z, x) \leq d(x, f(x))$. It follows by (gum4) that $d(z, f(z)) \leq d(x, f(x))$ and therefore that $B_{d(z,\, f(z))}(z) \subseteq B_{d(x,\, f(x))}(x)$ for all $B_{d(x,\, f(x))}(x) \in C$ by Lemma 2.1(1). Now, since f is strictly contracting on orbits, $d(f(z), f^2(z)) < d(z, f(z))$, and therefore $z \notin B_{d(f(z),\, f^2(z))}(f(z)) \subset B_{d(z,\, f(z))}(f(z))$. By Lemma 2.1(2), this is equivalent to $B_{d(f(z),\, f^2(z))}(f(z)) \subset B_{d(z,\, f(z))}(z)$, which is a contradiction to the maximality of $\mathcal{C}$. Thus, f has a fixed point.

Now let f be strictly contracting on X, and assume that x, y are two distinct fixed points of f. Then we get $d(x, y) = d(f(x), f(y)) < d(x, y)$, which is impossible. So, the fixed point of f is unique in this case. □

Note 2.1. We note here that uniqueness of fixed points, as in the theorem just given and earlier ones, is not usually something which can be derived from the Knaster–Tarski theorem or from Kleene's theorem.

2.2.4. *Generalized metrics and quasi-metrics in the context of multivalued mappings*

In [29], Khamsi, Kreinovich and Misane introduced a notion of generalized metric in order to study the stable model semantics of locally stratified programs; see Section 6. The notion of generalized metric defined in [29] is closely related to that of generalized ultra-metric introduced in Section 2.2.2, at least in the case which concerns us here, and we discuss this connection in this section.

By an *ordered semigroup V with identity* 0, we mean a semigroup V with identity 0 on which there is defined an ordering $\leq$ satisfying: $0 \leq v$ for all $v \in V$, and if $v_1 \leq v_2$ and $v_1' \leq v_2'$, then $v_1 + v_1' \leq v_2 + v_2'$.

DEFINITION 2.18. *Let V be an ordered abelian semigroup with identity* 0, *and let X be an arbitrary set. A* generalized metric *on X is a mapping* $d : X \times X \to V$ *which satisfies the usual metric axioms* (d1)–(d4); *that is, d satisfies the following axioms for all* $x, y, z \in X$.

(i) $d(x, y) = 0$ *if and only if* $x = y$.
(ii) $d(x, y) = d(y, x)$.
(iii) $d(x, z) \leq d(x, y) + d(y, z)$.

The pair (X, d) *consisting of a set X and a generalized metric d on X is called a* generalized metric space relative to V.

DEFINITION 2.19. *Let V be an ordered abelian semigroup with identity* 0. *Assume that* $\alpha \geq 1$ *is either a countable ordinal or* ω_1, *the first uncountable ordinal, and that* $\mathbf{v} = (v_\beta)_{\beta<\alpha}$ *is a decreasing family of elements of V. Finally, let X be a generalized metric space relative to V, and let* $(x_\beta)_{\beta<\alpha}$ *be a family of elements of X. Then:*

(i) (x_β) *is said to* $\mathbf{v}$-cluster to $x \in X$ *if, for all* β, *we have* $d(x_\beta, x) < v_\beta$ *whenever* $\beta < \alpha$;
(ii) (x_β) *is said to be* $\mathbf{v}$-Cauchy *if, for all* β *and* δ, *we have* $d(x_\beta, x_\delta) < v_\beta$ *whenever* $\beta < \delta < \alpha$;
(iii) *X is said to be* complete *if, for every* $\mathbf{v}$, *every* $\mathbf{v}$*-Cauchy family* $\mathbf{v}$*-clusters to some element of X*;
(iv) *a set* $Y \subseteq X$ *will be called* complete *if, for every* $\mathbf{v}$, *whenever a* $\mathbf{v}$*-Cauchy family consists of elements of Y, it* $\mathbf{v}$*-clusters to some element of Y*.

Remark 2.1. As in Section 2.2.2, let $\gamma > 1$ be an arbitrary countable ordinal, form the set Γ_γ with the ordering defined in Section 2.2.2 and denote $2^{-\gamma}$ by 0. Taking V as Γ_γ and the binary operation $+$ as before, that is, $u + v = \max\{u, v\}$, we obtain an ordered abelian semigroup V with identity 0. It will be convenient to write $1/2\ 2^{-\alpha}$ for $2^{-(\alpha+1)}$, but note that $1/2\ 2^{-\alpha}$ is not then being used with its meaning in Section 2.2.2. However, with a slight change of notation (which we will not trouble to make and will not cause confusion in so doing), on taking V as Γ_γ, a generalized metric is then a continuity function as in Definition 2.4.

Note 2.2. For the rest of this section, V will be taken to be Γ_γ as in Remark 2.1.

A mapping $T : X \to 2^X$ is called a *multivalued* 1/2*-contraction* if, for every $x \in X$, for every $y \in X$ and for every $a \in T(x)$, there exists $a\ b \in T(y)$ such that $d(a, b) \leq 1/2d(x, y)$.

The following theorem was established in [29].

THEOREM 2.11. *Let X be a complete generalized metric space, let T be a multivalued* 1/2*-contraction on X such that T (x) is not empty for some* $x \in X$ *and suppose that for every* $x \in X$ *the set T (x) is complete. Then T has a fixed point*.

Let (X, d) be a generalized metric space in the sense of Definition 2.18 with respect to V as given in Remark 2.1. Then it is easy to see that d is in fact a generalized ultra-metric in the sense of Section 2.2.2. However, to avoid confusion arising from overuse of the term 'generalized ultra-metric', we will refrain from employing this term to mean a generalized metric in the sense of Definition 2.18, which happens to satisfy axiom (gum4).

The next two results were established in [54].

PROPOSITION 2.1. *Let X be a complete generalized metric space with respect to V, where V is as defined in Remark* 2.1. *Then X is spherically complete in the sense of Section* 2.2.2.

PROPOSITION 2.2. *Let (X, d, V) be a spherically complete generalized ultra-metric space in the sense of Section* 2.2.2, *where V is as defined in Remark* 2.1. *Then X is complete in the sense of the present section*.

This means, by virtue of Theorem 2.5, that we can reformulate the assumptions in Theorem 2.11 and thereby obtain the following theorem, Theorem 2.12. In fact, our conclusion relative to the second statement in Theorem 2.12 is a special case of Theorem 2.6.

THEOREM 2.12. *Let X be a spherically complete generalized ultra-metric space* (*with respect to V*) *and let f be multivalued, non-empty and strictly contracting on X. Then either of the following conditions ensures the existence of a fixed point of f.*

(i) *The set* $\{d(x, y) \mid y \in f(x)\}$ *has a minimum in X for all* $x \in X$.
(ii) *The set* $f(x)$ *is spherically complete for each* $x \in X$.

Finally, to close this section and the first part of the paper, we mention a result involving quasi-metrics in the context of multivalued mappings, as follows. In addition to the results of Khamsi, Kreinovich and Misane already discussed above, these authors also established in [29] a version of the Banach contraction mapping theorem for multivalued mappings. In [30] a version of Kleene's theorem and a version of Theorem 2.2 were established for multivalued mappings.

The latter result achieved, using quasi-metrics, a unification for multivalued mappings of Kleene's theorem, as given in [30], and the Banach contraction mapping theorem of Khamsi *et al.* comparable with that obtained by Rutten, Smyth and others for single-valued mappings. The reader is referred to [30] for full details.

PART II: SOME APPLICATIONS OF GENERALIZED DISTANCE FUNCTIONS

We are now in a position to discuss the role of some of the theorems we have just described in the context of the theory of computation and in logic programming semantics in particular, and we proceed to do this next. We begin by presenting the minimum background needed in logic programming, concentrating on semantics and largely ignoring implementation and procedural matters. Our general reference for this subject is [17]; we refer to [63] for an account of the growth of logic programming and of its role as a major tool in various parts of computer science, such as database systems, artificial intelligence, natural language processing, machine learning and building expert systems etc.

3. LOGIC PROGRAMS

A logic programming system comprises four main facets: (i) the syntax or expressiveness of the system and its computational adequacy (relative, say, to SLDNF-resolution);[5] (ii) the procedural semantics of the system or what is output by the interpreter; (iii) the declarative semantics or logical meaning of the output; and (iv) the fixed-point semantics. These four issues are highly interconnected, and it is important that the three semantics just mentioned should coincide in some sense; see Theorem 3.1 for example. In fact, what is usually meant by the term *declarative semantics* is some natural model canonically associated with each program permitted by the syntax, and realized as the (least, minimal, unique etc.) fixed point of an operator determined by the program. The existence of such models is an advantage possessed by logic programs over conventional imperative or object-oriented programs in giving logic programs a clear, machine-independent meaning. Unfortunately, most systems with enhanced syntax permit many canonical models, and it is by no means obvious in general which of them best captures the intended meaning of the programmer, depending on his or her view of non-monotonic reasoning. Indeed, the study of these standard models, such as the well-founded model (van Gelder *et al*. [64]), the stable model (Gelfond and Lifschitz [65]) or the perfect model and the weakly perfect model (Przymusinski [19]), and of the corresponding operators, accounts for a high proportion of the research undertaken on the foundations of the subject. It should be noted that the canonical models just mentioned are in general different, and it is interesting to know when they are equal, for this provides conditions under which we have coincidence of the various ways of considering non-monotonic reasoning; we take up this point in Section 4.

[5]For example, the class of definite logic programs is computationally adequate relative to SLD-resolution; that is, it can compute all partial recursive functions. We note further that SLD-resolution is a standard implementation of logic programming and means Linear resolution with Selection function for Definite clauses. Similarly, SLDNF-resolution stands for SLD-resolution augmented with the Negation as Failure rule. We refer the reader to [17] for full details of these matters.

3.1. Syntax of normal logic programs

Given a first-order language $\mathcal{L}$, a *normal logic program P (with underlying language $\mathcal{L}$) is a finite set of clauses of the form*

$$\forall(A \leftarrow B_1 \wedge \cdots \wedge B_k \wedge \neg B_{k+1} \wedge \cdots \wedge \neg B_n),$$

usually written as

$$A \leftarrow B_1, \ldots, B_k, \neg B_{k+1}, \ldots, \neg B_n$$

or as

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

or more simply as

$$A \leftarrow \texttt{body},$$

where `body` denotes the conjunction $L_1 \wedge \cdots \wedge L_n$, usually written as $L_1, \ldots, L_n$. Here, A and all the B_i are atoms in $\mathcal{L}$, each L_i is a literal in $\mathcal{L}$ (an atom B_i or a negated atom $\neg B_i$), $\leftarrow$ denotes implication, and the universal quantifier is understood. The atom A is called the *head* of the clause, and each L_i is called a *body literal* of the clause. By an abuse of notation, we allow n to be zero or, in other words, we allow the body to be empty, in which case we are dealing with the *unit clause*, or *fact*, $A \leftarrow$. A program P is called *positive* or *definite* if no clause contains a negated atom.

EXAMPLE 5. *The following program (taken from [38]) computes the transitive closure of a graph.*

$$\begin{aligned} r(X, Y, E, V) &\leftarrow m([X, Y], E) \\ r(X, Z, E, V) &\leftarrow m([X, Y], E), \neg m(Y, V),\ r(Y, Z, E, [Y|V]) \\ m(X, [X|T]) &\leftarrow \\ m(X, [Y|T]) &\leftarrow m(X, T) \\ e(a) &\leftarrow \qquad \textit{for all } \alpha \in N \end{aligned}$$

Here, N denotes a finite set containing the nodes appearing in the graph as elements. In the program, uppercase letters denote variable symbols, lowercase letters constant symbols, and lists are written using square brackets as usual under Prolog. One evaluates a goal (the negation of the object one wishes to compute) such as $\leftarrow r(x, y, e, [x])$, where x and y are nodes and e is a graph specified by a list of pairs denoting its edges. The goal is supposed to succeed (or the interpreter outputs 'yes') when x and y can be connected by a path in the graph. The predicate m implements membership of a list. The last argument of the predicate r acts as an accumulator which collects the list of nodes which have already been visited in an attempt to reach y from x.

3.2. Semantics of normal logic programs

The usual approach to the declarative semantics of logic programs P is via Tarski's notions of interpretation and model, which are standard apparatus in mathematical logic. However, since we are at all times dealing with sets of clauses, Herbrand interpretations will suffice for our purposes; see [17, Chapter 1]. Thus, given a logic program P with underlying language $\mathcal{L}$, we form the Herbrand base B_P of P as defined in Example 2.3. Then a *two-valued interpretation* or simply an *interpretation* for P is a mapping from B_P to the classical, or two-valued, truth set {*true*, *false*}. Such an interpretation gives a truth value to each ground atom in $\mathcal{L}$ and extends, in the usual way, to give truth value to any closed well-formed formula, including clauses. Moreover, each interpretation can be identified with the subset of B_P on which it takes the value *true*. Thus, the set I_P of all interpretations will be naturally identified with the power set of B_P; it therefore carries the structure of a complete lattice (and a domain) under the order of set inclusion. In particular, a *model* for P is an interpretation I for P such that all clauses in P evaluate to true in I. Of course, as already noted, models are of particular importance in studying the semantics of P. Since clauses are universally quantified, checking their truth relative to an interpretation amounts to checking the truth of all their ground instances in that interpretation. We denote the set of all ground instances of clauses in P by ground(P), and it is often this set that one works with, rather than with P, when discussing questions of a theoretical nature.

A *partial interpretation* or *three-valued interpretation* I is a mapping from B_P to the truth set {*true* (t), *false* (f), *undefined* (u)} and can be identified with a pair (I^+, I^-) of disjoint subsets of B_P. Given a partial interpretation $I = (I^+, I^-)$, atoms in I^+ carry the truth value *true* in I and atoms in I^- the value *false* in I. Atoms which are neither in I^+ nor in I^- carry the truth value *undefined*. Partial interpretations are interpreted in one of the standard three-valued logics such as Kleene's strong three-valued logic, which tells one how the undefined value, u, relates to the other truth values under conjunction, disjunction and negation; see [21, 22, 66]. Once this is done, a truth value can again be given to any ground formula in $\mathcal{L}$. A partial interpretation (I^+, I^-) is called *total* if $I^+ \cup I^- = B_P$, and such an interpretation can be naturally identified with an element of I_P. The set $I_{P,3}$ of all partial interpretations is a complete partial order, indeed complete semilattice, and a domain under the ordering: $(I_1^+, I_1^-) \leq (I_2^+, I_2^-)$ if and only if $I_1^+ \subseteq I_2^+$ and $I_1^- \subseteq I_2^-$, where we take the bottom element to be $\bot = (\emptyset, \emptyset)$. Total interpretations are in fact maximal elements in the ordering just given.

3.3. Some operators determined by logic programs

There are various operators associated with a logic program P. They map interpretations to interpretations, and their importance lies in the fact that the various canonical models for P can be realized as fixed points of one or other of them. We discuss two of these operators now and others later on. The first, and perhaps the most important, is the *immediate consequence operator*or *single-step operator* $T_P : I_P \to I_P$ due to Kowalski and van Emden, see [17, 67, 68], defined as follows: T_P (I) is the set of all $A \in B_P$ such that there is a ground instance $A \leftarrow L_1, \ldots, L_n$ of a clause in P with head A satisfying $I \models L_1 \wedge \cdots \wedge L_n$, where $I \models L_1 \wedge \cdots \wedge L_n$ means that $L_1 \wedge \cdots \wedge L_n$ is true in I.

The operator T_P has many important and pleasing properties, and we summarize some of these next. First, if P is definite, then T_P is continuous on I_P. Therefore, it has a least fixed point lfp (T_P) given by Kleene's theorem. Moreover, one has the following theorem due to Apt, Kowalski and van Emden, see [67, 68] which, amongst other things, gives a form of the Gödel completeness theorem relating soundness and completeness for definite logic programming systems, see also [17].

THEOREM 3.1. *For any definite logic program P, we have* lfp $(T_P) = T_P \uparrow \omega = \{A \in B_P \mid P \vdash A\} = \{A \in B_P \mid P \models A\} = M_P$.

Thus, provability ($\vdash$) from P of a ground atom A relative to SLD-resolution coincides with it being a logical consequence ($\models$) of P, and both coincide with truth relative to the *least Herbrand model* M_P, which is the intersection of all Herbrand models for P. Moreover, because of continuity, the iterates T_P^n of T_P close off at ω, which gives us the means, in principle, of finding M_P. For these reasons, M_P is, for definite programs P, usually taken to be the standard model for P or, in other words, the programmer's intended meaning for P, as mentioned earlier.

Next, for any normal logic program, whether definite or not, T_P has the property that an interpretation I is a (two-valued) model for P precisely when T_P $(I) \subseteq I$ or, in other words, precisely when I is a pre-fixed point of T_P. The fixed points of T_P are of particular importance since they are the *supported models* or models for the *Clark completion of* P; see [17, 69]. It is argued in [18] that they are the appropriate models to consider, since an atom A belongs to such a model M if and only if there is a clause $A \leftarrow$ `body` in ground

(P) with `body` true in M, and hence the program itself supports the belief that A is true in M. Thus, the *supported model semantics* or *Clark completion semantics* is important, and it can be argued that it is 'the' standard model for P or the model best able to capture the intended meaning of P.

Therefore, it can be further argued that the fixed points of T_P are fundamental in studying the semantics of logic programming systems. Yet a major problem arises: if P is not definite, then T_P is not monotonic as can easily be seen by considering the program with the two clauses $p(0) \leftarrow$ and $p(s(x)) \leftarrow \neg p\,(x)$, which computes the even natural numbers. Therefore, the Knaster–Tarski theorem is not in general applicable as a means of finding fixed points, and this is the primary reason for our interest in alternative methods, such as those based on generalized distance functions, for finding fixed points of (non-monotonic) operators.

The second operator we consider is due to Fitting [66] and is the three-valued operator Φ_P defined as a mapping on partial interpretations $K = (K^+, K^-)$ as follows. We set $\Phi_P\,(K) = (I^+, I^-)$, where I^+ is the set of all $A \in B_P$ with the property that there exists a clause $A \leftarrow$ `body` in ground (P) such that `body` is true in K, and I^- is the set of all $A \in B_P$ such that for all clauses $A \leftarrow$ `body` in ground (P) we have that `body` is false in K, truth and falsehood being taken here relative to a three-valued logic as mentioned earlier.

We note that Φ_P is always monotonic, but not necessarily continuous. Thus, the Knaster–Tarski theorem applies and shows the existence of a least fixed point of Φ_P, although we may really have to iterate into the transfinite to reach it in the absence of continuity. It was shown in [66], and in [21, 22, 38] for acceptable programs, how fixed points of Φ_P relate to those of T_P. We see later in Section 5 that the fixed-point theorems of [50, 52] can sometimes be applied to show uniqueness of the fixed points of Φ_P which, incidentally, cannot be shown by means of the Knaster–Tarski theorem, as already noted.

4. ACCEPTABLE AND Φ*-ACCESSIBLE PROGRAMS

The use of ultra-metrics in algebra and in logic is well-established; see, for example, [70, 71] for such applications within valuation theory and algebraic geometry. In the opposite direction, see [72] for some interesting decidability and model-theoretic results relating to ultra-metrics arising in the context of (i) fields with valuations and (ii) sets A^λ of mappings from ordinals λ to sets A. However, metric and ultra-metric methods were introduced to logic programming by Fitting in [27] in analysing the semantics of the acceptable programs of Apt and Pedreschi [38] (these programs are important in termination analysis). Therefore, the fixed-point theorem employed in [27] is the Banach contraction mapping theorem. This is rather restrictive in so much as it is often useful to make transfinite constructions and definitions, although these may well be shown later to close off at ω, as this is important for computability purposes. In [22, 61], the present authors, inspired by the properties of acceptable programs, defined certain classes of programs, called Φ-accessible and Φ^*-accessible programs, which have the property that each program in the class has a unique supported model, and showed that it follows from this property that all the different semantics mentioned in Section 3 in fact coincide. These latter classes of programs were defined in terms of various three-valued logics and include the acceptable programs and certain other important classes, and are also known to be computationally adequate; they therefore are interesting in providing a semantically unambiguous setting with enhanced syntax and full computational power.

The proof of the existence and uniqueness of the supported models we gave in [22] for the Φ-accessible and Φ^*-accessible programs was by means of three-valued logics. In this section, we give an alternative proof based on Theorem 2.10 thereby illustrating the use of d-gums in logic programming semantics.

The following definition is taken from [38], where it was employed in defining acceptable programs; we will use it here as the basis of the more general Φ^*-accessible programs.

DEFINITION 4.1. *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) 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.

By the term *level mapping for P*, we mean a function $l: B_P \to \gamma$, where γ is an arbitrary (countable) ordinal >1; given a level mapping l, we always assume that l has been extended to all literals by setting $l\,(\neg A) = l\,(A)$ for each $A \in B_P$. If $l\,(A) = n$, we say that *the level of A is n or* that *A has level n*. Level mappings have been used in logic programming in a variety of contexts including problems concerned with termination, and with completeness, and also to define (generalized) metrics; see [27, 31, 38, 45]. We will see in Section 5 how they can be used to define generalized ultra-metrics in the sense of Definition 2.12. However, one of their main uses is in providing syntactic conditions on programs under which a satisfactory standard model can be obtained, and an instance of this usage is given in the next definition.

DEFINITION 4.2. *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 for P^-, such that the following condition holds. For each clause $A \leftarrow L_1, \ldots, L_n$ in ground(P), we either have $I \models L_1 \wedge \cdots \wedge L_n$ and $l(A) > l(L_i)$ for all $i = 1, \ldots, n$ or there exists an $i \in \{1, \ldots, n\}$ such that $I \not\models L_i$ and $l(A) > l(L_i)$.*

The Φ^*-accessible programs are a common generalization of acyclic, locally hierarchical and acceptable programs; see [38, 73]. As already noted, the present authors gave a unified treatment in [22] of these classes of programs by means of operators in various three-valued logics.

For the remainder of this section, let P denote a Φ^*-accessible program which satisfies the defining conditions with respect to a model I and a level mapping $l : B_P \rightarrow \gamma$. As in Section 2, we let Γ_γ denote the set $\{2^{-\alpha} | \alpha \leq \gamma\}$ ordered by $2^{-\alpha} < 2^{-\beta}$ if and only if $\beta < \alpha$, and here denote $2^{-\gamma}$ by 0.

For $J, K \in I_P$, we now define $d(K, K) = 0$, and $d(J, K) = 2^{-\alpha}$, where J and K differ on some atom $A \in B_P$ of level α, but agree on all ground atoms of lower level. It was shown in [25] that (I_P, d) is a spherically complete generalized ultra-metric space. For $K \in I_P$, we denote by K' the set K restricted to the predicate symbols in Neg_P^*. By analogy with [27], we now define for all $J, K \in I_P$: $d_1(J, K) = d(J', K')$ and $d_2(J, K) = d(J \backslash J', K \backslash K')$. Next, we define the function $f : I_P \rightarrow \Gamma$ by $f(K) = 0$ if $K \backslash K' \subseteq I$ and otherwise $f(K) = 2^{-\alpha}$, where α is the smallest ordinal such that there is an atom $A \in K \backslash K'$ with $l(A) = \alpha$ and $A \notin I$. Finally, we define $\varrho(J, K) = \max\{d_1(J, I), d_1(K, I), d_2(J, K), f(J), f(K)\}$ for all $J, K \in I_P$.

Proposition 4.1. *The space (I_P, ϱ) is a spherically complete d-ultra-metric.*

Proof. That (I_P, ϱ) is a d-ultra-metric space we leave to the reader. For spherical completeness, let $(\mathcal{B}_\alpha)$ be a (decreasing) chain of balls in I_P with centres I_α. Let K 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 ordinal β with $A \in I_\alpha$ for all $\alpha \geq \beta$. We show that for each ball $B_{2^{-\alpha}}(I_\alpha)$ in the chain we have $d(I_\alpha, I) \leq 2^{-\alpha}$, which suffices to show that K is in the intersection of the chain. Indeed, it is easy to see by the definition of ϱ that all I_β with $\beta > \alpha$ agree on all atoms of level less than α. Hence, by definition of K we obtain that K and I_α agree on all atoms of level less than α, as required. □

The next proposition is analogous to [27, Proposition 7.1].

Proposition 4.2. *Let P be Φ^*-accessible with respect to a level mapping l and a model I.* Then for all $J, K \in I_P$ with *$J \neq K$ we have $\varrho(T_P(J), T_P(K)) < \varrho(J, K)$. In particular, the following results hold.*

(i) $d_1(T_P(J), I) < d_1(J, I)$, *whenever* $d_1(J, I) \neq 0$, and $d_1(T_P(J), I) = 0$ *whenever* $d_1(J, I) = 0$.
(ii) $f(T_P(J)), f(T_P(K)) < \varrho(J, K)$.
(iii) $d_2(T_P(J), T_P(K)) < \varrho(J, K)$.

Proof. It suffices to prove properties (i)–(iii). For convenience, we identify Neg_P^* with the subset of B_P containing predicate symbols from Neg^*_P.

(i) First note that $d_1(T_P(J), I) = d_1(T_{P^-}(J), I)$ since d_1 only depends on the predicate symbols in Neg_P^*. Let $d(J, I) = 2^{-\alpha}$. We show that $d(T_{P^-}(J), I) \leq 2^{-(\alpha+1)}$. We know that J' and I' agree on all ground atoms of level less than α and differ on an atom of level α. It suffices to show now that $T_{P^-}(J)'$ and I' agree on all ground atoms of level $\leq \alpha$.

Let A be a ground atom in Neg_P^* with $l(A) \leq \alpha$ and suppose that $T_{P^-}(J)$ and I differ on A. Assume first that $A \in T_{P^-}(J)$ and $A \notin I$. Then there must be a ground instance $A \leftarrow L_1, \ldots, L_m$ of a clause in P^- such that $J \models L_1 \wedge \cdots \wedge L_m$. Since I is a fixed point of T_{P^-}, and using Definition 4.2. there must also be a k such that $I \not\models L_k$ and $l(L_k) < \alpha$. Note that the predicate symbol in L_k is contained in Neg_P^*. So we obtain $I \not\models L_k$, $J \models L_k$ and $l(L_k) < \alpha$, which is a contradiction to the assumption that J and I agree on all atoms in Neg_P^* of level less than α. Now assume that $A \in I$ and $A \notin T_{P^-}(J)$. It follows that there is a ground instance $A \leftarrow L_1, \ldots, L_m$ of a clause in P^- such that $I \models L_1 \wedge \cdots \wedge L_m$ and $l(A) > l(L_1), \ldots, l(L_m)$ by Definition 4.2.. But then $J \models L_1 \wedge \cdots \wedge L_m$ since J and I agree on all atoms of level less than α and consequently $A \in T_{P^-}(J)$. This contradiction establishes the first statement in (i). The second statement in (i) follows by a similar argument, noting that in this case $J' = I'$.

(ii) It suffices to show this for K. Assume $\varrho(J, K) = 2^{-\alpha}$. We show that $f(T_P(K)) \leq 2^{-(\alpha+1)}$, for which in turn we have to show that, for each $A \in T_P(K)$ not in Neg_P^* with $l(A) \leq \alpha$, we have $A \in I$. Assume that $A \notin I$ for such an A. Since $A \in T_P(K)$, there is a ground instance $A \leftarrow L_1, \ldots, L_m$ of a clause in P with $K \models L_1 \wedge \cdots \wedge L_m$. Since $A \notin I$, there must also be a k with $I \not\models L_k$ and $l(A) > l(L_k)$ by Definition 4.2. If the predicate symbol of L_k belongs to Neg_P^* then, since K and I agree on all atoms in Neg_P^* of level less than α, we obtain $K \not\models L_k$ which contradicts $K \models L_1 \wedge \cdots \wedge L_m$. If the predicate symbol in L_k does not belong to Neg_P^*, then L_k is an atom and since $f(K) \leq 2^{-\alpha}$ we obtain $I \models L_k$, which is again a contradiction.

(iii) Let $\varrho(J, K) = 2^{-\alpha}$, and let A be not in Neg_P^* with $l(A) \leq \alpha$ and $A \in T_P(J)$. By symmetry, it suffices to show that $A \in T_P(K)$. Since $A \in T_P(J)$, we must have a ground instance $A \leftarrow L_1, \ldots, L_m$ of a clause in P with $J \models L_1 \wedge \cdots \wedge L_m$. If $I \models L_1 \wedge \cdots \wedge L_m$, then $l(L_k) < l(A) \leq \alpha$ for all k, and since J and K agree on all atoms of level less than α we obtain $K \models L_1 \wedge \cdots \wedge L_m$, and hence $A \in T_P(K)$. If there is some L_k such that $I \nvDash L_k$, then without loss of generality $l(L_k) < l(A) \leq \alpha$ by Definition 4.2. Now, if the predicate symbol of L_k belongs to Neg_P^* then, since $d_1(J, I) \leq 2^{-\alpha}$, we obtain from $J \vDash L_k$ that $I \vDash L_k$, which is a contradiction. Also, if the predicate symbol of L_k does not belong to Neg_P^*, then L_k is an atom and since $f(J) \leq 2^{-\alpha}$, we obtain $I \vDash L_k$, again a contradiction. This establishes (iii) and completes the proof. □

We are now in a position to prove the main result of this section.

THEOREM 4.1. *Let P be a* Φ*-*accessible program. Then P has a unique supported model.*

Proof. By Proposition 4.2, T_P is strictly contracting with respect to ϱ, which in turn is a spherically complete d-ultra-metric by Proposition 4.1. So, by Theorem 2.10, the operator T_P must have a unique fixed point, yielding a unique supported model for P. □

5. LOCALLY STRATIFIED PROGRAMS

We now turn our attention to the class of locally stratified programs due to Przymusinski [19], beginning with their definition.

DEFINITION 5.1. *Let P be a normal logic program, let* $l: B_P \rightarrow \gamma$ *be a level mapping and let* $A \leftarrow A_1, \ldots, A_{k_l}, \neg B_1, \ldots \neg B_{l_l}$ *denote a typical clause in ground* (P). *Then P is called:*

(1) locally stratified (with respect to l) *if the inequalities* $l(A) \geq l(A_i)$ *and* $l(A) > l(B_j)$ *hold for all i and j in each clause in* ground (P),
(2) locally hierarchical (with respect to l) *if the inequalities* $l(A) > l(A_i)$, $l(B_j)$ *hold for all i and j in each clause in* ground (P).

Notice the use, again, of the level mapping involved in this definition as a syntactic device. This time, the stated conditions prevent 'negation through recursion', that is, they prevent an atom occurring in the head of a clause and simultaneously occurring negated in its body. It is this fact which permits the demonstration of the existence of a satisfactory standard model and also the derivation of its properties.

The locally stratified programs form one of the most important classes in logic programming and are in fact a generalization of the stratified programs defined by Apt, Blair and Walker in [18]. Przymusinski gave a non-constructive, and fairly involved, argument to show that each locally stratified program has a unique natural, supported model, known as the perfect model, preferable to any other model in a precise sense defined in [19]; constructive proofs of its existence and properties were given in [25, 74]. Of course, the locally hierarchical programs form a strict subclass of the locally stratified programs. Furthermore, it is known that many programs used in practice fall into the former class (of locally hierarchical programs), that each program in it has a unique supported model [25, 74] and that this class is computationally adequate, provided that the safe use of cuts is allowed [26]. We will sketch here how the fixed-point theory of these classes of programs can be treated by means of the theorems in Section 2.2.2, referring the reader to [54] for full details.

In order to proceed, we next cast I_P into a generalized ultrametric space. We do this by first viewing I_P as a domain, as in Example 2.3, and then forming the rank function given by Definition 2.15. Specifically, suppose that P is a logic program which is locally stratified with respect to the level mapping $l: B_P \rightarrow \gamma$, as in Definition 5.1. Then, as noted in Example 2.3, I_P can be thought of as a domain whose compact elements are the interpretations corresponding to the finite subsets of B_P. Now form the set Γ_γ as in Section 2.2.2, and define the rank function r_l *induced by* l by setting $r_l(I) = \max\{l(A); A \in I\}$ for every finite $I \neq \emptyset$ and take $r_l(\emptyset) = 0$. Denote the generalized ultra-metric resulting from r_l by d_l. Then, by Theorem 2.8, we see that (I_P, d_l) is a spherically complete generalized ultrametric space.

We now have the following result.

THEOREM 5.1. *Let P be a normal logic program which is locally stratified with respect to a level mapping l. Then P has a supported model. If, further, P is locally hierarchical with respect to l, then P has a unique supported model.*

Proof. It was shown in [53] that T_P is contracting since P is locally stratified, and that there cannot exist a ball $B_\pi(J)$ in (I_P, d_l) such that $d(I, T_P(I)) = \pi$ for all $I \in B_\pi(J)$. Therefore, it follows from Theorem 2.7 that T_P has a fixed point and hence that P has a supported model.

Next, if P is locally hierarchical, it was shown in [25] that T_P is strictly contracting. Therefore, by Theorem 2.7 again, it follows that T_P has a unique fixed point and so P has a unique supported model, as required. □

In the same way, the domain $I_{P,3}$ can be turned into a generalized ultra-metric space and we obtain a result corresponding to Theorem 5.1. In particular, we see that for locally hierarchical programs P, both Φ_P and the related operator Φ_P*, defined in [22], have a unique fixed point. Programs for which Φ_P* possesses a unique fixed point (the Φ*-accessible programs) have already been observed to be interesting and important insomuch as many of the standard models for them coincide, and therefore, for such programs, the various ways of viewing non-monotonic reasoning coincide. The locally hierarchical programs have this property and so, too, do the acceptable programs of [38]. Classes of programs with this property have elsewhere been called *unique supported model classes* by the authors, and characterized in [21, 22] in terms of the fixed points of Φ_P* in various three-valued logics. Theorem 5.1, or rather its (sketch) proof as given here, shows that generalized ultra-metric methods and Theorem 2.7 are powerful tools in carrying out investigations of this type.

6. THE STABLE MODEL SEMANTICS

We finally consider briefly, from our current point of view, the well-known and important stable model semantics of Gelfond and Lifschitz; see [65].

When studying non-monotonic reasoning and deductive databases, it is often convenient to consider extended disjunctive logic programs and to allow two different kinds of negation. One of these is interpreted as classical negation and the other is interpreted procedurally as negation as failure; see [17] for this notion. We introduce the following terminology following [29, 54, 65] closely.

Let $\mathcal{L}$ denote a first-order language. A literal L in $\mathcal{L}$ is called *ground* if it contains no variable symbols. We denote the set of all ground literals in $\mathcal{L}$ by $\texttt{Lit}$. A *rule* r in $\mathcal{L}$ is a universally quantifed expression of the following type:

$$L_1 \vee \cdots \vee L_n \leftarrow L_{n+1} \wedge \cdots \wedge L_m \wedge \texttt{not}\ L_{m+1} \wedge \cdots \wedge \texttt{not}\ L_k,$$

where each $L_i \in$ Lit. Given such a rule r, we define $\texttt{Head}(r) = \{L_1, \ldots, L_n\}$, $\texttt{Pos}(r) = \{L_{n+1}, \ldots, L_m\}$ and $\texttt{Neg}(r) = \{L_{m+1}, \ldots, L_k\}$. The keyword not may be interpreted as negation as failure. An (*extended disjunctive*) *program* Π is a set of (disjunctive) rules. The term 'extended' refers to the fact that two kinds of negation are employed, and the term 'disjunctive' refers to the appearance of more than a single literal in the heads of rules and to the disjunction between them. A normal logic program can therefore be understood as a special type of extended disjunctive program (in which '¬' is replaced by '$\texttt{not}$').

We note that a program is usually defined as a finite set of rules as above, but the literals L_i are allowed to be non-ground. However, as with a normal logic program, we can always replace a program by the set of all ground instances of its rules. This will yield an infinite set if function symbols are present, and a finite set otherwise (in which case Π is called an extended disjunctive database). Either way, in the sequel we assume that all the rules in an extended program are ground. Finally, a rule r, as above, will usually be written in the form

$$L_1, \ldots, L_n \leftarrow L_{n+1}, \ldots, L_m, \texttt{not}\ L_{m+1}, \ldots, \texttt{not}\ L_k.$$

Given a set Π of ground rules as just defined, it is possible to define a multivalued version T_Π of the single-step operator, to define supported models for Π, and to show that these coincide with the fixed points of T_Π, see [22]. Thus, fixed points of multivalued mappings and, consequently, corresponding fixed-point theorems enter very generally into the discussion. We shall not, however, pursue this line here in complete generality. Instead, we briefly consider another multivalued operator which encapsulates a view of non-monotonic reasoning due to Gelfond and Lifschitz. This leads to the well-known concept of *stable model*, and we show how its existence can be derived from Theorem 2.5.

In order to describe the stable model semantics or answer set semantics for programs, we first consider programs without negation, $\texttt{not}$. Thus, let Π denote a disjunctive program in which $\texttt{Neg}(r)$ is empty for each rule $r \in \Pi$. A subset X of $\texttt{Lit}$, that is, $X \in 2^{\texttt{Lit}}$, is said to be *closed by rules in* Π if, for every rule $r \in \Pi$ such that $\texttt{Pos}(r) \subseteq X$, we have that $\texttt{Head}(r) \cap X \neq \emptyset$. The set $X \in 2^{\texttt{Lit}}$ is called an *answer set* for Π if it is closed by rules in Π and satisfies the following conditions:

(1) if X contains complementary literals, then $X = \texttt{Lit}$, and
(2) X is minimal; that is, if $A \subseteq X$ and A is closed by rules of Π, then $A = X$.

We denote the set of answer sets of Π by $\alpha(\Pi)$.

Now suppose that Π is a disjunctive program that may contain $\texttt{not}$. For a set $X \in 2^{\texttt{Lit}}$, consider the program Π^X defined by:

(1) if $r \in \Pi$ is such that $\texttt{Neg}(r) \cap X$ is not empty, then we remove r, that is, $r \notin \Pi^X$, and
(2) if $r \in \Pi$ is such that $\texttt{Neg}(r) \cap X$ is empty, then the rule r' belongs to Π^X, where r' is defined by $\texttt{Head}(r') = \texttt{Head}(r)$, $\texttt{Pos}(r') = \texttt{Pos}(r)$ and $\texttt{Neg}(r') = \emptyset$.

It is clear that the program Π^X does not contain $\texttt{not}$ and therefore $\alpha(\Pi^X)$ is defined. Following Gelfond and Lifschitz [65], we define the operator GL : $2^{\texttt{Lit}} \to 2^{2^{\text{Lit}}}$ by GL(X) $= \alpha(\Pi^X)$. Finally, we say that X is an *answer set* or a *stable model* for Π if $X \in \alpha(\Pi^X)$, that is, if $X \in \text{GL}(X)$. In other words, X is an answer set for Π if it is a fixed point of the multivalued mapping GL. Again, we use the notation $\alpha(\Pi)$ for the set of answer sets of Π in the general case.

The following example will help to illustrate these ideas.

EXAMPLE 6.1. Take Π as follows:

$$p(0) \vee q(0) \leftarrow$$
$$p(a) \vee q(0) \leftarrow q(0) \wedge \texttt{not}\, p(0).$$

If X is any set of literals not containing $p(0)$, then Π^X is the program

$$p(0) \vee q(0) \leftarrow$$
$$p(a) \vee q(0) \leftarrow q(0),$$

and the answer sets of Π^X are $\{p(0)\}$ and $\{q(0)\}$. Thus, $\alpha(\Pi^X) = \{\{p(0)\},\{q(0)\}\}$. Since $X = \{q(0)\}$ is a suitable choice of X in that it does not contain $p(0)$, we see that $X \in \alpha(\Pi^X)$ and hence that $\{q(0)\}$ is an answer set for Π.

On the other hand, suppose that X is any set of literals which does contain $p(0)$. In this case, the program Π^X is as follows:

$$p(0) \vee q(0) \leftarrow .$$

Again, the only answer sets of Π^X are $\{p(0)\}$ and $\{q(0)\}$. Since $X = \{p(0)\}$ is a suitable choice of X in that it does contain $p(0)$ this time, we see that $\{p(0)\}$ is an answer set for Π, and indeed is the only one other than $\{q(0)\}$. Thus, $\alpha(\Pi) = \{\{p(0)\}, \{\mathrm{q}(0)\}\}$.

In this example, $\mathrm{GL}(X)$ contains the two elements $\{p(0)\}$ and $\{q(0)\}$ for any set X of literals, and hence is multivalued. Moreover, both $\{p(0)\}$ and $\{q(0)\}$ are fixed points of GL.

DEFINITION 6.1. *An extended disjunctive program* Π *is called* locally stratified *if there exists a mapping* (*a level mapping*) l : $\mathtt{Lit} \leftarrow \gamma$, *where* γ *is as usual a countable ordinal*, *such that for every* (*ground*) *rule* r *of* Π, *where* r *has the form*

$$L_1, \ldots, L_n \leftarrow L_{n+1}, \ldots, L_m, \mathtt{not}\, L_{m+1}, \ldots, \mathtt{not}\, L_k,$$

the following inequalities hold: $l(L) \geq l(L')$ *and* $l(L) > l(L'')$, *where* L, L' *and* L'' *denote, respectively, elements of* $\mathtt{Head}(r)$, $\mathtt{Pos}(r)$ *and* $\mathtt{Neg}(r)$.

This definition clearly generalizes Definition 5.1.

We close this section by showing that the existence of a stable model for a locally stratified extended disjunctive logic program Π follows from Proposition 2.1 and Theorem 2.12, and hence, ultimately, from Theorem 2.5. Thus, Theorem 2.5 gives a unified treatment of the fixed-point theory of locally stratified programs and extended disjunctive programs.

Proceeding along the lines of Definition 2.15, first let $\mathtt{Lit}_\alpha$ denote the set $\{L \in \mathtt{Lit} \mid l(L) = \alpha\}$, where l is the level mapping with respect to which Π is locally stratified. We now define a generalized metric d on $2^{\mathtt{Lit}}$ as follows: if $A = B$, then $d(A, B) = 0$; if $A \neq B$, then $d(A, B) = 2^{-\alpha}$, where α is the smallest ordinal for which $A \cap \mathtt{Lit}_\alpha \neq B \cap \mathtt{Lit}_\alpha$. The resulting generalized metric space turns out to be complete, as shown in [29]. It is straightforward to see that the GL operator satisfies the assumptions of Theorem 2.12. Therefore, GL has a fixed point which is a stable model of Π, as required.

7. SOME RECENT DEVELOPMENTS AND FURTHER WORK

In this section, we make brief comments on a number of recent and fairly recent applications of distance functions to the theory of computation, and give some pointers to the literature. In the main, these have not yet been discussed in the paper, other than to mention them in the Introduction, and they are, we believe, areas and applications where promising new results may be expected. In presenting them, we more or less follow the order in which we introduced the various distance functions we have considered.

First, metrics themselves have many more applications in semantics than we have specifically mentioned earlier; see the book [13] (or [75] for a summary) for applications to transition systems, and [76] for applications to Scott's information systems, for example. In a rather different direction, we refer to our own recent and ongoing work in relation to the integration of logic-based systems and neural networks, in which metrics play a vital role; see [77–79] for example. Furthermore, the ideas concerning level mappings and generalized ultra-metrics discussed here have been taken up in [80, 81] in further developing logic programming semantics itself.

As far as ultra-metrics are concerned, the references [6–8] (and the references within these works) contain a comprehensive account of a mathematical model of cognitive processes in which a key idea is that of the p-adic hierarchical tree-like space. The development employs ultra-metrics significantly in, for example, constructing information models over various tree-like structures corresponding to ultra-metric topologies. Similar ideas, namely, that there are natural tree-like structures defined by ultra-metric topologies are employed in the papers [9, 10]. One of the problems considered in the papers just cited is to characterize how well time series data can be embedded in an ultra-metric topology, and this has applications in a number of areas including unique fingerprinting of a time series. Finally, we mention some applications of ultra-metrics in bioinformatics in the papers [11, 12]. In particular, in [11], basic properties of p-adic numbers are used in a new approach to describing the main aspects of DNA sequences and the genetic code. A central role in this is played by an ultra-metric p-adic information space whose basic elements are nucleotides, codons and genes. It is shown that genetic code degeneracy is related to the p-adic distance between codons. It is clear that these all represent especially interesting application areas for ultra-metric methods in information processing in which yet more, important results may be expected.

Turning to quasi-metrics, we mention a number of applications, as follows. (i) First, in [82], complexity spaces were introduced in order to study complexity analysis of programs. These spaces are quasi-metric spaces and have been extensively examined in [82, 83] and later papers; in particular, in [83] it is shown that O'Neill's conjecture [59] on the relationship between norms and valuations holds in the context of the theory of semivaluation spaces. It turns out also that the weightable quasi-metrics (or partial metrics) are important in this context, and this fact relates complexity analysis and denotational semantics nicely. (ii) In [84], quasi-metrics are used to define abstract interpretations of programs in static analysis in the sense of Cousot and Cousot. The value d (x,

y) encodes not only the fact of approximation between x and y, but also the error introduced by the approximation. For this reason, d is a quasi-metric, but not a metric: if x approximates y, it does not follow that y approximates x. In this framework, Theorem 2.2 is used in place of the Knaster–Tarski theorem. (iii) In [85], the authors consider the problem of estimating the probability of accessing objects in replicated databases in order to minimize overload (and problems related to conflicts and consistency in the databases) in accessing a given object x in the database. This is done by introducing a simple 'probability of access' $v(x)$ intended to estimate the probability that object x will be accessed in a time interval $[0, T]$. It is shown, empirically, that $v(x)$ is a good estimator and also that an appropriate mathematical framework for the theory is that of a quasi-metric lattice. The latter has the structure of both a quasi-metric and a lattice satisfying the properties: $d(x \vee z, y \vee z) \leq d(x, y)$, and $d(x \wedge z, y \wedge z) \leq d(x, y)$ for all x, y, z, where $\vee$ and $\wedge$ denote the lattice operations. (iv) Another application of 'quasi-metrics' to programming languages is given in [86], but note that the use of the term quasi-metric in [86] is stronger than ours (axioms (d1), (d2) and (d4) of Section 2.1 are assumed in [86]). Here, the authors consider the space of all finite and infinite words over an alphabet. They consider metrics and (balanced)[6] quasi-metrics d, in their sense, defined on this space and closely related to the well-known Baire metric; see [16]. They establish a fixed-point theorem for mappings f on X, which satisfy a contraction property on some orbit,[7] and apply it to discuss the average-case analysis of probabilistic divide and conquer algorithms.

As already noted, the relationship between generalized distance functions of various types and (Scott–Ershov) domains, both viewed as abstract models of computation, has been explored in considerable depth. This is especially so in relation to attempts to reconcile these two concepts. We mention next two recent developments in this theme, the first to be found in [87, 88] and the second in [89].

In [87, 88], partial metrics are shown to be related to the so-called Martin measurements [90]. The latter were introduced as a quantitative means of capturing the degree of *indefiniteness* of elements in a Scott–Ershov domain considered as objects arising within a computational approximation. Several correspondences between partial metrics, measurements on domains and properties of the respective spaces are established in [87, 88]. In the same vein, we refer to the papers [91–93] in which it is shown, respectively, that all domains can be equipped with a partial metric (obtained independently in [88]); that partial metrics can be interpreted as a non-trivial generalization of Birkhoff's notion of a valuation on a lattice to a semivaluation on a semilattice; and that many of the important constructions of Matthews for partial metrics hold for the more general class of partial quasi-metrics.

In [89], generalized ultra-metrics are examined in relation to domains. The starting point of these investigations is the *formal-balls model* of [47], which was introduced as a means of capturing properties of metric spaces by domain-theoretic methods, including a proof of the Banach contraction mapping theorem (Theorem 2.3) by applying Kleene's theorem (Page 2). In [23], this approach is employed to give a proof of Theorem 2.4 (for f strictly contracting) by means of the Knaster–Tarski theorem (Page 2 also). In [89], the resulting correspondence between domains and generalized ultra-metrics is investigated in category-theoretic terms, and it turns out that many properties of a generalized ultra-metric can be characterized by conditions on its formal-balls model. Furthermore, two modified fixed-point theorems related to Theorems 2.3 and 2.4 are established in [89]. Nevertheless, the work of [89] casts considerable doubt on the extent to which these methods can be used for connecting generalized ultra-metrics and domain theory, and indeed shows that they are somewhat limited. Despite this, we have been able to utilize essentially these ideas in a satisfactory way, as can be seen in the appropriate sections of this paper. The general question of how one may unify generalized ultra-metrics and domains, however, remains open.

Finally, in a direction related to those in the previous two or three paragraphs, Michael Bukatin argues in his PhD thesis, [94], for an approach to software engineering based on continuity and approximation and, in particular, continuity in (constructive and 'continuous') mathematics; somewhat similar thinking (the use of continuous mathematics) underlies the probabilistic semantics presented in [95]. Bukatin and J. S. Scott take this idea further in [96] where they propose to measure the distance between programs. Their framework is that of domain theory, and they show that a suitable distance function for measuring the distance between two programs must be a *relaxed metric*, a notion very close to partial metrics (again, the axiom $d(x, x) = 0$ is dropped) and close to the valuations of O'Neill [59]. Preliminary results in a similar theory for logic programs were obtained in [97], based on the results of [95].

8. CONCLUSIONS

Our discussion shows that there is a considerable number of different and interesting generalized distance functions which have significant use within the theory of computation. In relation to the theory of programming languages, it is true that the majority of developments and applications (of denotational semantics, for example) are expressed in terms of order theory and are therefore qualitative. Nevertheless, the results discussed here show that the quantitative approach provided by distance functions can be viewed

[6]See [40] for the definition of this term.

[7]Specifically, there is a point $x \in X$ and $c \in (0,1)$ such that for all $n \in \mathbb{N}$, we have $d(f^{n+1}(x), f^n(x)) \leq cd(f^n(x), f^{n-1}(x))$.

as complementary and orthogonal to the qualitative approach and indeed is sometimes indispensible, just as it is in mathematics and computer science in general.

ACKNOWLEDGEMENTS

The authors wish to thank Professors Branko Dragovich, Andrei Khrennikov and Fionn Murtagh for their invitation to contribute to this special edition of the Computer Journal. They also thank three anonymous referees whose detailed and insightful comments, suggested references and corrections of some technical errors considerably enhanced the contents and presentation of this paper. Finally, they thank Michel Schellekens for several useful references to recent work in the subject matter of the paper, which had escaped their attention.

FUNDING

The first author acknowledges partial support by the Boole Centre for Research in Informatics (BCRI) at University College Cork. The second author acknowledges partial support by the German Federal Ministry of Education and Research (BMBF) under the SmartWeb project (grant 01 IMD01 B), and by the Deutsche Forschungsgemeinschaft (DFG) under the ReaSem project.

REFERENCES

[1] Smyth, M.B. (1992) Topology. In Abramsky, S., Gabbay, D.M. and Maibaum, T.S. (eds), *Handbook of Logic in Computer Science*, Vol. 1, pp. 641–761. Oxford University Press, Oxford, UK.

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

[3] Kopperman, R. (1988) All topologies come from generalized metrics. *Amer. Math. Monthly*, **95**, pp. 89–97.

[4] Arnold, A. and Nivat, M. (1980) The metric space of infinite trees: algebraic and topological properties. *Fund. Inform.*, **3**, pp. 445–476.

[5] Arnold, A. and Nivat, M. (1980) Metric interpretations of infinite trees and semantics of non-deterministic recursive programs. *Theoret. Comput. Sci.*, **11**, pp. 181–205.

[6] Albeverio, S., Khrennikov, A. and Kloeden, P. (1999) Memory retrieval as a *p*-adic dynamical system. *Biosystems*, **49**, pp. 105–115.

[7] Khrennikov, A. (1998) Human subconscious as the p-adic dynamical system. *J. Theoret. Biology*, **193**, pp. 179–196.

[8] Khrennikov, A. (2004) Information Dynamics in Cognitive, Psychological, Social and Anomalous Phenomena, Fundamental Theories of Physics, Vol. 138. Kluwer Academic Publishers, Dordrecht, Boston, London.

[9] Murtagh, F. (2004) On ultrametricity, data coding, and computation. *J. Classification*, **21**, pp. 167–184.

[10] Murtagh, F. (2005) Identifying the ultrametricity of time series. *Eur. Phys. J.*, **43**, pp. 573–579.

[11] Dragovich, B. and Dragovich, A. (2006). A p-dic model of DNA sequence and genetic code. http://www.arxiv.org/abs/q-bio.GN/0607018.

[12] Khrennikov, A. and Kozyrev, S.V. (2007) Genetic codenext term on the diadic plane. *Phys. A: Statist. Mech. Appl.*, **381**, pp. 265–272.

[13] de Bakker, J. and de Vink, E. (1996) Control Flow Semantics. *Foundations of Computing Series*. The MIT Press, Cambridge, Massachusetts.

[14] Reed, G.M., Roscoe, A.W. and Wachter, R.F. (eds) (1991) *Topology and Category Theory in Computer Science*. Oxford University Press, Oxford, UK.

[15] Flax, L. (1994) Using the Hamming Distance Between Logical Formulae to Examine Damage to an Artificial Neural Network. In Bozşahin, C., Halici, U., Oflazer, K. and Yalabik, N. (eds), *Proc. 3rd Turkish Symp. Artificial Intelligence and Neural Networks*, Ankara, Turkey, June, pp. 307–319. Middle East Technical University.

[16] Stoltenberg-Hansen, V., Lindström, I. and Griffor, E.R. (1994) Mathematical Theory of Domains. *Cambridge Tracts in Theoretical Computer Science*, Vol. 22. Cambridge University Press, Cambridge, UK.

[17] Lloyd, J.W. (1987) *Foundations of Logic Programming*, (2nd edn). Springer, Berlin.

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

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

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

[21] Hitzler, P. and Seda, A.K. (1999) Acceptable Programs Revisited. In Etalle, S. and Smaus, J.-G. (eds), *Proc. Workshop on Verification in Logic Programming, 16th Int. Conf. Logic Programming, ICLP'99*, Las Cruces, New Mexico, November. Electronic Notes in Theoretical Computer Science, Vol. 30, pp. 59–76. Elsevier Science Publishers.

[22] Hitzler, P. and Seda, A.K. (1999) Characterizations of Classes of Programs by Three-valued Operators. In Gelfond, M., Leone, N. and Pfeifer, G. (eds), *Logic Programming, Non-monotonic Reasoning Proc. 5th Int. Conf. Logic Programming and Non-monotonic Reasoning, LPNMR'99*, El

Paso, Texas, USA, December. Lecture Notes in Artificial Intelligence, Vol. 1730, pp. 357–371. Springer, Berlin.

[23] Hitzler, P. and Seda, A.K. (2003) Generalized metrics and uniquely determined logic programs. *Theoret. Comput. Sci.*, **305**, 187–219. *Proc. Dagstuhl Seminar 00231*, Topology in Computer Science: Constructivity; Asymmetry and Partiality; Digitalization, Schloss Dagstuhl, Germany, June, 2000.

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

[25] Seda, A.K. and Hitzler, P. (1997) Topology and Iterates in Computational Logic. In Flagg, R., Hart, K., Norden, J., Tymchatyn, E. and Tuncali, M. (eds), *Proc. 12th Summer Conf. General Topology and its Applications*, Ontario, Canada, August. Topology Proceedings, Vol. 22, 1999, pp. 427–469. Auburn University, USA.

[26] Seda, A.K. and Hitzler, P. (1998) Strictly Level-Decreasing Logic Programs. In Butterfield, A. and Flynn, S. (eds), *Proc. Second Irish Workshop on Formal Methods, IWFM'98*, NUI, Cork Ireland, July Electronic Worshops in Computing (eWiC), 1999, pp. 1–18. British Computer Society, Swindon, UK.

[27] Fitting, M. (1994) Metric methods: Three examples and a theorem. *J. Log. Program.*, **21**, pp. 113–127.

[28] Hitzler, P. and Seda, A.K. (1999) Multivalued Mappings, Fixed-Point Theorems and Disjunctive Databases. In Butterfield, A. and Flynn, S. (eds), *Proc. 3rd Irish Workshop on Formal Methods, IWFM'99*, NUI, Galway, Ireland, July. Electronic Workshops in Computing (eWiC), pp. 1–18. British Computer Society, Swindon, UK.

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

[30] Hitzler, P. and Seda, A.K. (1999) Some Issues Concerning Fixed Points in Computational Logic: Quasimetrics, Multivalued Mappings and the Knaster-Tarski Theorem. In Comfort, W.W., Heckmann, R., Kopperman, R.D. and Narici, L. (eds), *Proc. 14th Summer Conf. General Topology and its Applications*, Long Island, USA, August. Topology Proceedings, Vol. 24, pp. 223–250. Auburn University, USA.

[31] Seda, A.K. (1997) Quasi-metrics and the semantics of logic programs. *Fund. Inf.*, **29**, pp. 97–117.

[32] Davey, B.A. and Priestley, H.A. (2002) *Introduction to Lattices and Order* (2nd edn). Cambridge University Press, Cambridge, UK.

[33] Ganter, B. and Wille, R. (1999) *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin.

[34] Nachbin, L. (1965) Topology and Order, *Mathematical Studies*, Vol. 4. Van Nostrand, Princeton, NJ.

[35] Joslyn, C. and Bruno, W.J. (2005) Weighted Pseudo Distances for Categorization in Semantic Hierarchies. In Dau, F., Mugnier, M.-L. and Stumme, G. (eds), *Conceptual Structures: Common Semantics for Sharing Knowledge*, *Proc. 13th Int. Conf. Conceptual Structures, ICCS 2005*, Kassel, Germany, July. Lecture Notes in Computer Science, Vol. 3596, pp. 381–395. Springer, Berlin.

[36] Hitzler, P. (2004) Default Reasoning over Domains and Concept Hierarchies. In Biundo, S., Frühwirth, T. and Palm, G. (eds), *KI2004: Advances in Artificial Intelligence*. *Proc. 27th Annual German Conf. AI, KI2004*, Ulm, Germany, September. Lecture Notes in Artificial Intelligence, Vol. 3238, pp. 351–365. Springer.

[37] Hitzler, P. and Krötzsch, M. (2006) Querying Formal Contexts with Answer Set Programs. In Schärfe, H., Hitzler, P. and Øhrstrøm, P. (eds), *Conceptual Structures: Inspiration and Application*, *Proc. 14th International Conf. Conceptual Structures, ICCS 2006*, Aalborg, Denmark, July 2006. Lecture Notes in Artificial Intelligence, Vol. 4068, pp. 413–426. Springer.

[38] Apt, K.R. and Pedreschi, D. (1993) Reasoning about termination of pure Prolog programs. *Inf. Comput.*, **106**, pp. 109–157.

[39] Flagg, B. and Kopperman, R. (1997) Continuity spaces: Reconciling domains and metric spaces. *Theoret. Comput. Sci.*, **177**, pp. 111–138.

[40] Künzi, H.-P.A. (2001) Non-Symmetric Distances and Their Associated Topologies: About the Origins of Basic Ideas in the Area of Asymmetric Topology. In Aull, C. and Lowen, R. (eds), *Handbook of the History of General Topology*, Vol. 3, pp. 853–968. Kluwer, Dordrecht.

[41] Willard, S. (1970) *General Topology*. Addison-Wesley, Reading, MA.

[42] Scott, D.S. (1982) Domains for denotational semantics. In Nielsen, M. and Schmidt, E.M. (eds), *Automata, Languages and Programming, 9th Colloq., Proc.*, Aarhus, Denmark, mJuly. Lecture Notes in Computer Science, Vol. 140, pp. 577–613. Springer.

[43] Bonsangue, M.M., Breugel, F. and Rutten, J.J.M.M. (1995) Generalized Ultrametric Spaces: Completion, Topology, and Powerdomains via the Yoneda Embedding. Technical report. Centrum voor Wiskunde en Informatica (CWI), Amsterdam, The Netherlands.

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

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

[46] Baier, C. and Majster-Cederbaum, M.E. (1997) Metric semantics from partial order semantics. *Acta Inform.*, **34**, pp. 701–735.

[47] Edalat, A. and Heckmann, R. (1998) A computational model for metric spaces. *Theoret. Comput. Sci.*, **193**, pp. 53–73.

[48] Rutten, J.J.M.M. (1996) Elements of generalized ultrametric domain theory. *Theoret. Comput. Sci.*, **170**, pp. 349–381. Revised version of [49].

[49] Rutten, J.J.M.M. (1995) Elements of Generalized Ultrametric Domain Theory. Technical Report CS-R9507. CWI, Stichting Mathematisch Centrum, Amsterdam, The Netherlands.

[50] Priess-Crampe, S. and Ribenboim, P. (2000) Ultrametric spaces and logic programming. *J. Log. Program.*, **42**, pp. 59–70.

[51] Priess-Crampe, S. and Ribenboim, P. (2000) Logic programming and ultrametric spaces. *Rend. Math.*, **VII**, pp. 1–13.

[52] Priess-Crampe, S. and Ribenboim, P. (1993) Fixed points, combs and generalized power series. *Abh. Math. Sem. Univ. Hamburg.*, **63**, pp. 227–244.

[53] Bouamama, S., Misane, D. and Priess-Crampe, S. (2000) An Application of Ultrametric Spaces in Logic Programming. *Proc. Sixth Maghrebian Conf. Computer Sciences, MCSEAI'2000*, Fès, Morocco, November 1–3, pp. 1–5. Maghrebian Information Processing Society.

[54] Hitzler, P. and Seda, A.K. (2002) The Fixed Point Theorems of Priess-Crampe and Ribenboim in Logic Programming. In Kuhlmann, F.V., Kuhlmann, S. and Marshall, M. (eds), *Valuation Theory and its Applications*, *Proc. International Valuation Theory Conf.*, University of Saskatchewan in Saskatoon, Canada, July 1999. Fields Institute Communications Series, Vol. 32, pp. 219–235. American Mathematical Society, Providence, RI.

[55] Matthews, S.G. (1986) Metric Domains for Completeness. rTechnical Report 76. Department of Computer Science, University of Warwick, Warwick. PhD Thesis, 1985.

[56] Matthews, S.G. (1994) Partial Metric Topology. In Andima, S., Itzkowitz, G., Kong, Y., Kopperman, R., Misra, P., Narici, L. and Todd, A. (eds), *Papers on General Topology and Applications: Eighth Summer Conf. General Topology and its Applications*, Queens College, CUNY, NY, June. Annals of the New York Academy of Sciences, Vol. 728, pp. 183–197. New York Academy of Sciences.

[57] Kahn, G. (1974) The Semantics of a Simple Language for Parallel Processing. *Proc. IFIP Congress 74*, Stockholm, August, pp. 471–475. Elsevier Science Publishers.

[58] Heckmann, R. (1999) Approximation of metric spaces by partial metric spaces. *Appl. Categ. Structures*. **7**, pp. 71–83.

[59] O'Neill, S.J. (1996) Partial Metrics, Valuations, and Domain Theory. In Andima, S., Flagg, R., Itzkowitz, G., Kong, Y., Kopperman, R. and Misra, P. (eds), *Papers on General Topology and Applications: Eleventh Summer Conf. General Topology and Applications*, University of Southern Maine, ME, August. Annals of the New York Academy of Sciences. pp. 304–315. New York Academy of Sciences.

[60] Matthews, S.G. (1992) The Cycle Contraction Mapping Theorem. Technical Report 228. Department of Computer Science, University of Warwick, Warwick.

[61] Hitzler, P. and Seda, A.K. (2000) Dislocated topologies. *J. Electr. Eng.*, **51**, pp. 3–7. *Proc. 2nd Slovakian Conf. Applied Mathematics*, Bratislava. Slovak Academy of Sciences.

[62] Sarma, R.D. (2007) Relational Topology. Preprint, Department of Mathematics, University of Delhi, India.

[63] Apt, K.R., Marek, V.W., Truszczýnski, M. and Warren, D.S. (eds) (1999) *The Logic Programming Paradigm: A 25-Year Perspective*. Springer, Berlin.

[64] van Gelder, A., Ross, K.A. and Schlipf, J.S. (1991) The well-founded semantics for general logic programs. *J. ACM*, **38**, pp. 620–650.

[65] Gelfond, M. and Lifschitz, V. (1991) Classical negation in logic programs and disjunctive databases. *New Gener. Comput.* **9**, pp. 365–385.

[66] Fitting, M. (1985) A Kripke–Kleene semantics for general logic programs. *J. Log. Program.*, **2**, pp. 295–312.

[67] Apt, K.R. and van Emden, M.H. (1982) Contributions to the theory of logic programming. *JACM*, **29**, pp. 841–862.

[68] Kowalski, R.A. (1974) Predicate Logic as a Programming Language. In Rosenfeld, J. (ed.), *Proc. IFIP'74*, Stockholm, Sweden, August, pp. 569–574. North-Holland, Amsterdam.

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

[70] Kuhlmann, F., Kuhlmann, S. and Marshall, M. (eds) (2002) *Valuation Theory and its Applications*, *Proc. Int. Valuation Theory Conf.*, University of Saskatchewan in Saskatoon, Canada, July, 1999. Fields Institute Communications Series, Vol. 32. American Mathematical Society, Providence, RI.

[71] Ribenboim, P. (1996) The new theory of ultrametric spaces. *Period. Math. Hungar.*, **32**, pp. 103–111.

[72] Delon, F. (1984) Espaces ultramétriques. *J. Symbolic Logic*, **49**, pp. 405–424.

[73] Cavedon, L. (1989) Continuity, Consistency, and Completeness Properties for Logic Programs. In Levi, G. and Martelli, M. (eds), *Proc. 6th Int. Conf. Logic Programming*, June, Portugal, pp. 571–584. MIT Press, Cambridge, MA.

[74] Cavedon, L. (1991) Acyclic programs and the completeness of SLDNF-resolution. *Theoret. Comput. Sci.*, **86**, pp. 81–92.

[75] de Bakker, J. and de Vink, E. (1996) A Metric Approach to Control Flow Semantics. In Andima, S., Flagg, R., Itzkowitz, G., Kong, Y., Kopperman, R. and Misra, P. (eds), *Papers on General Topology and Applications: Eleventh Summer Conference on General Topology and Applications,* University of Southern Maine, ME, August. Annals of the New York Academy of Sciences, pp. 11–27. New York Academy of Sciences.

[76] Edalat, A. and Smyth, M.B. (1992) Compact Metric Information Systems. In de Bakker, J., de Roever, W.-P. and Rozenberg, G. (eds), *Semantics: Foundations and Applications*. *Proc. REX Workshop*, Beekbergen, The Netherlands, June. Lecture Notes in Computer Science, Vol. 666, pp. 154–173. Springer, Berlin.

[77] Bader, S., Hitzler, P. and Hölldobler, S. (2006) The integration of connectionism and first-order knowledge representation and reasoning as a challenge for artificial intelligence. *Information*, **9**. Invited paper.

[78] Hitzler, P., Hölldobler, S. and Seda, A.K. (2004) Logic programs and connectionist networks. *J. Appl. Logic*, **2**, pp. 245–272.

[79] Seda, A.K. (2006) On the Integration of Connectionist and Logic-Based Systems. In Hurley, T., an Airchinnigh, M.M., Schellekens, M., Seda, A.K. and Strong, G. (eds), *Proc. 4th Irish Conf. on the Mathematical Foundations of Computer Science and Information Technology (MFCSIT'2004),* Trinity College Dublin, Ireland, July. Electronic Notes in Theoretical Computer Science (ENTCS), Vol. 161, pp. 109–130. Elsevier Science Publishers.

[80] Hitzler, P. (2005) Towards a systematic account of different semantics for logic programs. *J. Logic Comput.*, **15**, pp. 391–404.

[81] Hitzler, P. and Wendt, M. (2005) A uniform approach to logic programming semantics. *Theory Pract. Log. Program.*, **5**, pp. 123–159.

[82] Schellekens, M. (1995) The Smyth Completion: A Common Foundation for Denotational Semantics and Complexity Analysis. In Brookes, S., Main, M., Melton, A. and Mislove, M.(eds), *Proc. MFPS 11,* Tulane University, New Orleans, USA, March, Electronic Notes in Theoretical Computer Science, Vol. 1, pp. 211–232. Elsevier Science Publishers.

[83] Romaguera, S. and Schellekens, M. (2003) Norm-Weightable Riesz Spaces and the Dual Complexity Space. In Hurley, T., an Airchinnigh, M.M., Schellekens, M. and Seda, A.K. (eds), *Proc. 2nd Irish Conf. Mathematical Foundations of Computer Science and Information Technology (MFCSIT'2002)*, Galway, Ireland, July. Electronic Notes in Theoretical Computer Science, Vol. 74, pp. 1–17. Elsevier Science Publishers.

[84] Crazzolara, F. (1997) Quasi-Metric Spaces as Domains for Abstract Interpretation. In Falaschi, M., Navarro, M. and Policriti, A. (eds), *Proc. 1997 Joint Conf. Declarative Programming, APPIA-GULP-PRODE'97*, Grado, Italy, June, pp. 45–56. University of Udine, Italy.

[85] Castro-Company, F., Romaguera, S., Sánchez-Álvarez, J. and Tirado, P. (2007) A Quasi-Metric Lattice Approach for Access Prediction in Replicated Database Protocols. Technical Report. Instituto de Matemática Pura y Aplicada, Universidad Politécnica de Valencia, Spain.

[86] Rodríguez-López, J., Romaguera, S. and Valero, O. (2007) Denotational semantics for programming languages, balanced quasi-metrics and fixed points. *Int. J. Comput. Math.*, to appear.

[87] Waszkiewicz, P. (2002) Quantitative Continuous Domains. PhD Thesis, School of Computer Science, The University of Birmingham Edgbaston, Birmingham B15 2TT, UK.

[88] Waszkiewicz, P. (2003) Quantitative continuous domains. *Appl. Categ. Struct.*, **11**, pp. 41–67.

[89] Krötzsch, M. (2006) Generalized ultrametric spaces in quantitative domain theory. *Theoret. Comput. Sci.*, **368**, pp. 30–49.

[90] Martin, K. (2000) A Foundation for Computation. PhD Thesis, Department of Mathematics, Tulane University New Orleans, LA 70118, USA.

[91] Schellekens, M. (2003) A characterization of partial metrizability: Domains are quantifiable. *Theoret. Comput. Sci.*, **305**, pp. 409–432.

[92] Schellekens, M. (2004) The correspondence between partial metrics and semi-valuations. *Theoret. Comput. Sci.*, **315**, pp. 135–149.

[93] Künzi, H.-P.A., Pajoohesh, H. and Schellekens, M. (2006) Partial quasi-metrics. *Theoret. Comput. Sci.*, **365**, pp. 237–246.

[94] Bukatin, M.A. (2002) Mathematics of Domains. PhD Thesis, Brandeis University Waltham, MA, USA.

[95] Kozen, D. (1981) Semantics of probabilistic programs. *J. Comput. Syst. Sci.*, **22**, pp. 328–350.

[96] Bukatin, M.A. and Scott, J.S. (1997) Towards Computing Distances Between Programs via Scott Domains. In Adian, S. and Nerode, A. (eds), *Logical Foundations of Computer Science, 4th Int. Symp., LFCS'97 Proc.*, Yaroslavl, Russia, July. Lecture Notes in Computer Science, Vol. 1234, pp. 33–43. Springer, London.

[97] Seda, A.K. and Lane, M. (2003) On Continuous Models of Computation: Towards Computing the Distance Between (Logic) Programs. In Morris, J., Aziz, B. and Oehl, F. (eds), *Proc. 6th Int. Workshop in Formal Methods (IWFM'03),* Dublin City University, Dublin, Ireland, July. Electronic Workshops in Computing (eWiC), pp. 1–15. British Computer Society, Swindon.