GENERALIZED ULTRAMETRICS, DOMAINS AND AN APPLICATION TO COMPUTATIONAL LOGIC[1]

Anthony Karel Seda & Pascal Hitzler

1. Introduction

Fixed points of functions and operators are of fundamental importance in programming language semantics, in giving meaning to recursive definitions and to constructs which involve self-reference. It follows, therefore, that fixed-point theorems are also of fundamental importance in theoretical computer science. Often, order-theoretic arguments are available, in which case the well-known Knaster-Tarski theorem can be used to obtain fixed points. Sometimes, however, analytical arguments are needed involving the Banach contraction mapping theorem, as is the case, for example, in studying concurrency and communicating systems. Situations arise also in computational logic in the presence of negation which force non-monotonicity of the operators involved. A successful attempt was made in [5] to employ metrics and the contraction mapping theorem in studying some problematic logic programs. These ideas were taken further in [16] in examining quasi-metrics and in [17,18] in considering elementary ideas from topological dynamics in this same context of computational logic.

One thing which emerged from [17] was an application of a fixed-point theorem due to Sibylla Priess-Crampe and Paulo Ribenboim, see [10]. This theorem utilizes ultrametrics which

[1]This article is a slightly expanded version of a paper of the same title presented by the first named author at the 11th September meeting of the IMS, University of Ulster at Coleraine, September 1998.

are allowed to take values in an arbitrary partially ordered set and is a substitute for the contraction mapping theorem. The inspiration for this result appears to have come from applications within algebra and, in particular, to ordered abelian groups, and rings of generalized power series. However, as already indicated, our interest in it resides in its potential applications to theoretical computer science.

Our purpose in this note is to give some weight to the previous sentence by sketching the application we made in [17] of Theorem 1. Thus, in §2 we briefly consider generalized ultrametrics i.e. ultrametrics which take values in an arbitrary partially ordered set (not just in the non-negative reals) and state the fixed-point theorem of Priess-Crampe & Ribenboim, Theorem 1. In §3, we consider a natural way of endowing Scott domains with generalized ultrametrics. This step provides a technical tool which we need in §4 in applying Theorem 1 to finding fixed points of non-monotonic operators arising out of logic programs and deductive databases and hence to finding models for these.

2. Generalized ultrametric spaces: the fixed-point theorem of Priess-Crampe & Ribenboim

It will be convenient to give some basic definitions in this section, and to introduce some notation all of which is to be found in [10,11].

Definition 1 (Priess-Crampe & Ribenboim) 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 ultrametric space (gum)* if $d : X \times X \to \Gamma$ is a function satisfying the following conditions for all $x, y, z \in X$ and $\gamma \in \Gamma$:
(1) $d(x, y) = 0$ if and only if $x = y$;
(2) $d(x, y) = d(y, x)$;
(3) if $d(x, y) \leq \gamma$ and $d(y, z) \leq \gamma$, then $d(x, z) \leq \gamma$.

Of course, this definition is entirely standard except that the function d takes its values in the set Γ rather than in the set of non-negative real numbers, and to that extent is considerably more general. Moreover, as in the classical case, one can define "balls" in the context of generalized ultrametric spaces: for $0 \neq \gamma \in \Gamma$ and $x \in X$, the set $B_\gamma(x) = \{y \in X; d(x,y) \leq \gamma\}$ is called a γ-*ball* or just a *ball* in X. One then has the following elementary facts, see [10].

Fact 1 (1) If $\alpha \leq \beta$ and $x \in B_\beta(y)$, then $B_\alpha(x) \subseteq B_\beta(y)$. Hence every point of a ball is also its centre.
(2) If $B_\alpha(x) \subset B_\beta(y)$, then $\beta \not\leq \alpha$ (i.e. $\alpha < \beta$ if Γ is totally ordered).

A substitute in the present context is needed for the usual notion of completeness in (ultra)metric spaces, and this is provided by the notion of "spherical completeness" as follows. A generalized ultrametric space X is called *spherically complete* if $\bigcap \mathcal{C} \neq \emptyset$ for any chain $\mathcal{C}$ of balls in X. (By a "chain of balls" we mean, of course, a set of balls which is totally ordered by inclusion.)

A typical example, see [11], of a generalized ultrametric space is provided by the following function space in which the distance between two functions is the *set* of points on which they differ, and therefore is *not* numerical in nature.

Example 1 Take a non-empty set A and a set E with at least two elements. Let $H = \prod_{a \in A} E$ and define $d : H \times H \to \mathcal{P}(A)$ by $d(f,g) = \{a \in A; f(a) \neq g(a)\}$, where $\mathcal{P}(A)$ denotes the power set of A. Then $(H, d, \mathcal{P}(A))$ is a spherically complete gum.

A function $f : X \to X$ is called *strictly contracting* if $d(f(x), f(y)) < d(x,y)$ for all $x, y \in X$ with $x \neq y$. The following theorem, which is to be found in [10], can be thought of as an analogue of the Banach contraction mapping theorem.

Theorem 1 (Priess-Crampe & Ribenboim) *Let (X,d) be a spherically complete generalized ultrametric space and let $f : X \to X$ be strictly contracting. Then f has a unique fixed point.*

In fact, there are more general versions of this theorem for both single and multi-valued mappings, see [11]. As already noted, it is our belief that this theorem has a significant rôle to play in theoretical computer science in the study of the semantics of logic-based programming languages. Indeed, some applications in this area have been made in [11], and we discuss another one here in § 4.

3. Domains as GUMS

Domains are a special type of ordered set, as defined below. They were introduced independently by D. S. Scott and Y. L. Ershov as a means of providing structures for modelling computation, and to provide spaces to support the denotational semantics approach to understanding programming languages, see [20]. Usually, domains are endowed with the Scott topology, which is one of the T_0 (but not T_1) topologies of interest in theoretical computer science. However, under certain conditions, to be examined below, domains can be endowed with the structure of a generalized ultrametric space. This is not something normally considered in domain theory but, as we shall see, has interesting applications to the semantics of logic programs.

Let $(D, \sqsubseteq)$ denote a Scott domain with set D_C of compact elements, see [20]. Thus:

- $(D, \sqsubseteq)$ is a partially ordered set which, in fact, forms a complete partial order (cpo). Hence, D has a bottom element $\perp$, and the supremum $\sup A$ exists for all directed subsets A of D.
- The elements $a \in D_C$ satisfy: whenever A is directed and $a \sqsubseteq \sup A$, then $a \sqsubseteq x$ for some $x \in A$.
- For each $x \in D$, the set $\operatorname{approx}(x) = \{a \in D_C; a \sqsubseteq x\}$ is directed and $x = \sup \operatorname{approx}(x)$.
- If the set $\{a, b\} \subseteq D_C$ is consistent (there exists $x \in D$ such that $a \sqsubseteq x$ and $b \sqsubseteq x$), then $\sup\{a, b\}$ exists in D.

Several important facts emerge from these conditions including the existence (indeed construction) of fixed points of continuous functions, and the existence of function spaces (the category of domains is cartesian closed). Moreover, the compact elements provide an abstract notion of computability.

Example 2 (i) $(\mathcal{P}(N), \subseteq)$ is a domain whose compact elements are the finite subsets of N.
(ii) The set of all partial functions from N^n into N ordered by graph inclusion is a domain whose compact elements are the finite functions.

As already noted, domains carry a natural and important topology called the Scott topology. Under certain conditions the

Scott topology can be generated by a quasi-metric, see [16,19], but is never metrizable. However, by means of a construction similar to that discussed in [19], we can endow a domain with a generalized ultrametric, quite separate from its Scott topology, and this we discuss next.

Let γ denote an arbitrary countable ordinal i.e. one of the transfinite sequence $0, 1, 2, \ldots, \omega, \omega+1, \omega+2, \ldots, \omega 2, \omega 2+1, \omega 2+2, \ldots, \omega\omega, \omega\omega+1, \omega\omega+2, \ldots$. Let Γ_γ denote the set $\{2^{-\alpha}; \alpha < \gamma\}$ of symbols $2^{-\alpha}$ which we order by $2^{-\alpha} < 2^{-\beta}$ if and only if $\beta < \alpha$.

Definition 2 Let $r : D_C \to \gamma$ be a function, called a *rank function*, form $\Gamma_{\gamma+1}$ and denote $2^{-\gamma}$ by 0. Define $d_r : D \times D \to \Gamma_{\gamma+1}$ by $d_r(x,y) = \inf\{2^{-\alpha}; c \sqsubseteq x$ if and only if $c \sqsubseteq y$ for every $c \in D_C$ with $r(c) < \alpha\}$.

Then (D, d_r) is a generalized ultrametric space said to be *induced by* r. Moreover, (D, d_r) is spherically complete provided we impose one standing condition (SC) on the rank function r: for each $x \in D$ and for each ordinal $\alpha < \gamma$, the set $\{c \in \mathrm{approx}(x); r(c) < \alpha\}$ is directed whenever it is non-empty.

Theorem 2 *Under the standing condition (SC) on r, (D, d_r) is spherically complete.*

Full details of these results can be found in [17]. However, the key to obtaining Theorem 2 is the following lemma whose proof we sketch here; a key point in the details is that any point of a ball in a gum is its centre (Fact 1). To simplify notation denote the ball $B_{2^{-\alpha}}(x)$ by $B_\alpha(x)$.

Lemma 1 *Suppose that r satisfies condition SC, and let $B_\alpha(x) \subseteq B_\beta(y)$. Then the following hold.*
(1) $\{c \in \mathrm{approx}(x); r(c) < \beta\} = \{c \in \mathrm{approx}(y); r(c) < \beta\}$.
(2) $B_\alpha = \sup\{c \in \mathrm{approx}(x); r(c) < \alpha\}$ *and* $B_\beta = \sup\{c \in \mathrm{approx}(y); r(c) < \beta\}$ *both exist.*
(3) $B_\beta \sqsubseteq B_\alpha$.
Proof. Since $x \in B_\alpha(x)$, we have $x \in B_\beta(y)$ and hence $d_r(x,y) \leq 2^{-\beta}$. So (1) follows immediately from the definition of d_r.
Since $\{c \in \mathrm{approx}(x); r(c) < \beta\}$ is bounded by x, we get (2) from the consistent completeness of D, see [20].
For the third statement:

Step 1. Suppose $B_\alpha(x) \subset B_\beta(y)$. Then $\beta < \alpha$ by Fact 1 since Γ_γ is totally ordered. Thus $B_\beta = \sup\{c \in \text{approx}(y); r(c) < \beta\} = \sup\{c \in \text{approx}(x); r(c) < \beta\} \sqsubseteq \sup\{c \in \text{approx}(x); r(c) < \alpha\} = B_\alpha$, and so $B_\beta \sqsubseteq B_\alpha$ as required.
Step 2. Now suppose that $B_\alpha(x) = B_\beta(y) = B$, say.
Subcase 1. If $\alpha = \beta$, then it is immediate that $B_\alpha = B_\beta$.
Subcase 2. Suppose finally that $\alpha \neq \beta$ and suppose in fact that $\alpha < \beta$, so that $B_\alpha \sqsubseteq B_\beta$, with a similar argument if it is the case that $\beta < \alpha$. We show again that $B_\alpha = B_\beta$, and it suffices to obtain $d_r(B_\alpha, B_\beta) = 0$. By definition of d_r, B_α and B_β, we see that B_α and B_β are both elements of the ball B in question. Suppose that $d_r(B_\alpha, B_\beta) \neq 0$. Then there is a compact element c_1 such that the statement "$c_1 \sqsubseteq B_\alpha$ iff $c_1 \sqsubseteq B_\beta$" is false. Since $B_\alpha \sqsubseteq B_\beta$, it must be the case that $c_1 \not\sqsubseteq B_\alpha$ and $c_1 \sqsubseteq B_\beta$. By Fact 1 any point of a ball is its centre, and so we can take y to be B_β in the equation established in (1). We therefore obtain $B_\beta = \sup\{c \in \text{approx}(B_\beta); r(c) < \beta\}$. If $\{c \in \text{approx}(B_\beta); r(c) < \beta\}$ is empty, then B_α and B_β are both equal to the bottom element $\bot$ of D and we are done; so suppose $\{c \in \text{approx}(B_\beta); r(c) < \beta\} \neq \emptyset$. Since $c_1 \sqsubseteq B_\beta$, there is, by the condition SC, a compact element c_2 with $r(c_2) < \beta$ such that $c_1 \sqsubseteq c_2 \sqsubseteq B_\beta$. But then $c_2 \not\sqsubseteq B_\alpha$ otherwise we would have $c_1 \sqsubseteq c_2$ and $c_2 \sqsubseteq B_\alpha$ leading to the contradiction $c_1 \sqsubseteq B_\alpha$. But now we have a compact element c_2 with $r(c_2) < \beta$ and for which $c_2 \not\sqsubseteq B_\alpha$ and $c_2 \sqsubseteq B_\beta$, and this contradicts the fact that $d_r(B_\alpha, B_\beta) \leq 2^{-\beta}$. Hence, $B_\alpha = B_\beta$ as required. ■

4. Applications to Computational Logic

Conventional logic programming is concerned with computation as deduction (using SLD-resolution) from (possibly infinite) sets P of clauses of type

$$C_1 \vee \ldots \vee C_j \leftarrow A_1 \wedge \cdots \wedge A_{k_1} \wedge \neg B_1 \wedge \cdots \wedge \neg B_{l_1}$$

(for disjunctive databases) or of type

$$C \leftarrow A_1 \wedge \cdots \wedge A_{k_1} \wedge \neg B_1 \wedge \cdots \wedge \neg B_{l_1}$$

(for programs), where all the A's, B's and C's are atoms in some first order language $\mathcal{L}$, see [8] for details. A central problem in the theory is to give a canonical meaning (semantics) to P, and the standard solution of this problem is to find the fixed points of an operator T_P determined by P. (This compares with the problem of giving semantics to recursive definitions or to constructs involving self-reference in conventional programming languages. In both cases, the meaning is taken to be a fixed point of a function (or functor) which naturally arises from within the problem.)

For programs, we proceed as follows: form the set B_P of all ground (variable-free) atoms in $\mathcal{L}$ and its power set $I_P = \mathcal{P}(B_P)$ ordered by set inclusion (elements I of I_P can be naturally identified with interpretations, including the models, for P). Then $T_P : I_P \to I_P$ is defined by setting $T_P(I)$ to be the set of all ground atoms C in B_P for which there is a ground instance $C \leftarrow A_1 \wedge \cdots \wedge A_{k_1} \wedge \neg B_1 \wedge \cdots \wedge \neg B_{l_1}$[2] of a clause in P satisfying $I \models A_1 \wedge \cdots \wedge A_{k_1} \wedge \neg B_1 \wedge \cdots \wedge \neg B_{l_1}$. Some standard facts concerning T_P are as follows:

(a) If P contains no negation symbols (P is positive), then T_P is monotone (even continuous) and its least fixed point can be found by applying the Knaster-Tarski theorem (the fixed-point theorem for cpos) and gives a satisfactory semantics for P.

(b) If P contains negation symbols, then T_P is non-monotonic and we face the difficulty of finding fixed points of non-monotonic operators.

Note 1 There are various ways of considering T_P from the point of view of a dynamical system, the main issue being to control the evolution of the iterates $T_P^n(\emptyset)$ or more generally of $T_P^n(I)$ for some $I \in I_P$:

(i) Identify I_P with a product of two-point spaces endowed with the product of the discrete topologies (Cantor space) and then T_P can be thought of as a kind of shift operator; this relates to the work of Christopher Moore in [9], see also [21].

---

[2] A *ground instance* of a clause in P is an instance $C \leftarrow A_1 \wedge \cdots \wedge A_{k_1} \wedge \neg B_1 \wedge \cdots \wedge \neg B_{l_1}$ of a program clause in which each of the atoms C, A_i, B_j is an element of B_P i.e. a clause resulting from a program clause by assigning all the variable symbols to ground terms.

(ii) T_P can be thought of as a mapping on a closed subspace of the Vietoris space of B_P and hence as a set dynamical system, see [4,18].

For databases there are further problems in that the appropriate operator T is multi-valued and we want I such that $I \in T(I)$ (a fixed point of T). We shall not, however, discuss databases as such in detail, but instead refer the reader to [7] where a multi-valued version of the contraction mapping theorem can be found, and also an application of it to finding models of disjunctive databases.

Returning to programs, various syntactic conditions, see [1,2,12,13,14], have been considered in attempting to find fixed points of non-monotonic operators, including the following which is one of the most important:

Definition 3 Let $l : B_P \to \gamma$ be a mapping (a level mapping[3]) where γ is a countable ordinal. Call P:
(1) *Locally stratified with respect to l* (Przymusinski) if the inequalities $l(C) \geq l(A_i)$ and $l(C) > l(B_j)$ hold for all i and j in each ground instance of each clause in P.
(2) *Strictly level-decreasing with respect to* l, as in [17,18], if the inequalities $l(C) > l(A_i), l(B_j)$ hold for all i and j in each ground instance of each clause in P.

It is known that the class in (1) has several minimal, supported[4] models (due to Przymusinski, Gelfond, Lifschitz et al.) for each program in the class. Indeed, it is not a priori clear which of these models can be taken to be the natural semantics for any given program in class (1), and the choice depends on how one attempts to model non-monotonic reasoning. However, subclass (2) of (1) is interesting in that it is one of the rather

[3]Level mappings are used in logic programming in a variety of contexts including problems concerned with termination, and with completeness and also to define metrics, see [2,3,5].

[4]An interpretation I for P is said to be *supported* if $I \subseteq T_P(I)$. Such interpretations are important in logic programming, and this point is discussed in [1]. Since an interpretation I is a model for P iff $T_P(I) \subseteq I$, it follows that a model for P is supported iff it is a fixed point of T_P.

rare classes of programs which satisfy both of the following two properties (I) and (II) simultaneously, unlike the class (1) which obviously satisfies (I) but not (II):

(I) It is computationally adequate i.e. any partial recursive (computable) function can be computed by some program in class (2), see [18].

(II) For each program in (2) all the "natural" models coincide – so there is no argument about which is best. In fact, this statement is an improvement on the results obtained by Przymusinski in [12,13,14].

The statement (II) can be established by an application of the ideas discussed earlier by viewing I_P as a domain whose set of compact elements is the set I_C of all finite subsets of B_P, and we now indicate briefly how this is done.

Definition 4 Let $l : B_P \to \gamma$ be a level mapping. Define the rank function r_l *induced* by l by setting $r_l(I) = \max\{l(A); A \in I\}$ for every $I \in I_C$, with I non-empty, and taking $r_l(\emptyset) = 0$. Denote the generalized ultrametric resulting from r_l by d_l.

The following theorem was established in [17], and we note that the condition SC imposed on r (concerning directedness) is trivially satisfied by r_l.

Theorem 3 *Let P be strictly level-decreasing with respect to a level mapping l. Then T_P is strictly contracting with respect to the generalized ultrametric d_l induced by l.*

It follows from Theorems 1, 2 and 3 that T_P has a unique fixed point and therefore that P has a unique supported model. In turn, it follows that all the standard semantics for P coincide with the perfect model semantics (due to Przymusinski) which is the unique minimal supported model for P.

The interested reader can find full details of all the results discussed in this section in [17,18], and we close with a couple of simple examples of programs which do not compute anything in particular but which illustrate how level mappings arise, taking values in ordinals beyond ω.

Example 3 (1) Let P be the program consisting of the following three clauses:

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

$$p(o) \leftarrow$$

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

Define $l : B_P \to \omega + 1$ by $l(p(s^n(o))) = n$ and $l(q(s^n(o))) = \omega$ for all $n \in N$. Then P is strictly level-decreasing, and the unique supported model given by Theorem 3 is the set $\{p(s^{2n}(o)); n \in N\}$.
(2) This time take P to be as follows:

$$p(o, o) \leftarrow$$

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

$$p(y, s(x)) \leftarrow \neg p(y, x)$$

Define $l : B_P \to \omega\omega$ by $l(p(s^k(o), s^j(o))) = \omega k + j$, where ωk denotes the k^{th} limit ordinal. Then P is strictly level-decreasing and its unique supported model is $\{p(o, s^{2n}(o)); n \in N\} \cup \{p(s^{n+1}(o), s^{2k+1}(o)); k, n \in N\}$.

Example 4 Take the "even numbers" program:

$$p(o) \leftarrow$$

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

with the ω-level mapping l defined by $l(p(s^n(o))) = n$. Theorem 3 applies to this program and the set $\{p(o), p(s^2(o)), p(s^4(o)), \ldots\}$ of even numbers is the resulting unique fixed point of T_P.

Example 5 Consider the following program P:

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

$$p(x) \leftarrow r(x)$$

$$r(x) \leftarrow p(x)$$

$$q(o) \leftarrow$$

The set $\{q(o), p(s^n(o)), r(s^n(o))\}$ is a fixed point of T_P for every n. Therefore, T_P can never satisfy the hypothesis of Theorem 3. In fact, this program is locally stratified, but is never strictly level-decreasing for any level mapping because of the cycle created by the second and third clauses. Such a cycle would be prohibited in a strictly level-decreasing program, and this example shows that a locally stratified program need not have a contractive immediate consequence operator. ■

References

[1] K. R. Apt, H. A. Blair and A. Walker, *Towards a theory of declarative knowledge*, *in* Foundations of Deductive Databases and Logic Programming, (ed.) J. Minker, 89–148. Morgan Kaufmann Publishers Inc.: Los Altos, 1988.

[2] K. R. Apt and D. Pedreschi, *Reasoning about termination of pure prolog programs*, Information and Computation **106** (1) (1993), 109–157.

[3] M. Bezem, *Characterizing termination of logic programs with level mappings*, *in* Proceedings of the North American Conference on Logic Programming, (eds.) E. L. Lusk and R. A. Overbeek, 69–80. MIT Press, 1989.

[4] A. Edalat, *Dynamical systems, measures and fractals via domain theory*, Information and Computation **120** (1) (1995), 32–48.

[5] M. Fitting, *Metric methods: three examples and a theorem*, J. Logic Programming **21** (3) (1994), 113–127.

[6] P. Hitzler, Topology and Logic Programming Semantics. Diplomarbeit in Mathematik, Universität Tübingen, 1998.

[7] M. A. Khamsi, V. Kreinovich and D. Misane, *A new method of proving the existence of answer sets for disjunctive logic programs: a metric fixed point theorem for multi-valued mappings*, *in* Proceedings of the Workshop on Logic Programming with Incomplete Information, Vancouver, B.C., Canada, October 1993, pp. 58–73.

[8] J. W. Lloyd, Foundations of Logic Programming (second edition). Springer-Verlag: Berlin, 1988.

[9] C. Moore, *Generalized shifts: unpredictability and undecidability in dynamical systems*, Nonlinearity **4** (1991), 199–230.

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

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

[12] T. Przymusinski, *On the declarative semantics of deductive databases and logic programs*, *in* Foundations of Deductive Databases and Logic Programming, (ed.) J. Minker, 193–216. Morgan Kaufmann Publishers Inc.: Los Altos, 1988.

[13] T. Przymusinski, *Well-founded and stationary models of logic programs*, Annals of Mathematics and Artificial Intelligence **12** (1994), 141–187.

[14] T. Przymusinski, *Semantics of normal and disjunctive logic programs: a unifying framework*, *in* Proceedings of the Workshop on Non-Monotonic Extensions of Logic Programming at the 11th International Logic Programming Conference, ICLP' 95, Santa Margherita Ligure, Italy, June 1994. Springer-Verlag, 1995, pp. 43–67.

[15] A. K. Seda, *Topology and the semantics of logic programs*, Fundamenta Informaticae **24** (4) (1995), 359–386.

[16] A. K. Seda, *Quasi-metrics and the semantics of logic programs*, Fundamenta Informaticae **29** (1) (1997), 97–117.

[17] A. K. Seda and P. Hitzler, *Topology and iterates in computational logic*, Topology Proceedings Vol. 22, Proceedings of the 12th Summer Conference on Topology and its Applications: Special Session on Topology in Computer Science, Ontario, August 1997, pp. 1–43, to appear.

[18] A. K. Seda and P. Hitzler, *Strictly level-decreasing logic programs*, *in* Proceedings of the Second Irish Workshop on Formal Methods, (eds.) A. Butterfield and S. Flynn, Electronic Workshops in Computing, Springer-Verlag, 1998, pp. 1–18, to appear.

[19] M. B. Smyth, *Totally bounded spaces and compact ordered spaces as domains of computation*, *in* Topology and Category Theory in Com-

puter Science, (eds.) G. M. Reed, A. W. Roscoe and R. F. Wachter. Oxford University Press: Oxford, 1991.

[20] V. Stoltenberg-Hansen, I. Lindström and E. R. Griffor, Mathematical Theory of Domains. Cambridge Tracts in Theoretical Computer Science No. 22, Cambridge University Press: Cambridge, 1994.

[21] J. Walsh, Topology, Fractals and Domain Theory — Towards a Geometria. M.Sc. Thesis, University of Dublin, Trinity College, 1997.

A. K. Seda
Department of Mathematics
University College
Cork
Ireland
email:aks@bureau.ucc.ie

P. Hitzler
Mathematische Fakultät
Universität Tübingen
Germany
email:scmt6001@bureau.ucc.ie