Journal of ELECTRICAL ENGINEERING, VOL. 51, NO. 12/s, 2000, 54–58

UNIQUENESS OF THE FIXED POINTS OF SINGLE–STEP OPERATORS DETERMINED BY BELNAP'S[1] FOUR–VALUED LOGIC

Eleanor Clifford — Anthony Karel Seda *

Recently, Hitzler and Seda showed how a domain-theoretic proof can be given of the fact that, for a locally hierarchical program, the single-step operator T_P, defined in two-valued logic, has a unique fixed point. Their approach employed a construction which turned a Scott-Ershov domain into a generalized ultrametric space. Finally, a fixed-point theorem of Priess-Crampe and Ribenboim was applied to T_P to establish the result. In this paper, we extend these methods and results to the corresponding well-known single-step operators Φ_P and Ψ_P determined by P and defined, respectively, in three-valued and four-valued logics.

K e y w o r d s: Fixed point, operator, many-valued logic, Scott-Ershov domain, locally hierarchical program.

2000 *Mathematics Subject Classification*: Primary 68N17, Secondary 03B70, 03G10

1 INTRODUCTION

A common approach to giving meaning or "semantics" to programming language constructs is to assign an operator to the construct and look for its fixed points. In this approach, one often finds that the operator in question is monotonic and defined on a complete lattice or complete partial order (cpo), so that the well-known Knaster-Tarski theorem can be applied to obtain the required fixed points. However, in the case of logic programs P, see [9], the presence of negation (which enhances syntax and expressibility) leads to non-monotonicity of the usual single-step operator T_P associated with P and hence to inapplicability of the Knaster-Tarski theorem.

Various ways of overcoming this problem have been proposed in the literature, including the use of analytical and topological methods, see for example [12] and its references. Another approach, see [3,4], is to consider other operators such as Φ_P and Ψ_P defined in three-valued and four-valued logics. In particular, Fitting in [3,4] has drawn attention to an operator Ψ_P defined on the space $I_{P,4}$ of $\mathcal{FOUR}$-valued interpretations, or valuations, of the underlying first order language $\mathcal{L}$ of P, where $\mathcal{FOUR}$ denotes the four-valued logic due to Belnap [1] as employed by Fitting in [4]. Indeed, $\mathcal{FOUR}$ includes conventional two-valued and Kleene's strong three-valued logic, and others, as sublogics. In fact, $I_{P,4}$ carries two natural orderings $\leq_k$ and $\leq_t$. Under the first of these, Ψ_P extends the operator Φ_P defined over Kleene's strong three-valued logic and is monotonic; under the second, Ψ_P extends the operator T_P defined over two-valued logic, and is only monotonic when P is definite (does not contain negation). Thus, any result about Ψ_P pertains to T_P and Φ_P so that $I_{P,4}$ and Ψ_P provide a very convenient setting to study logic programming semantics in great generality.

The foregoing remarks raise several questions concerning the fixed points of T_P, Φ_P, Ψ_P and their interaction. Many of these questions have been pursued in [4], see also the references there, and they will be discussed briefly here in Section 2. The tool usually employed to obtain fixed points is the Knaster-Tarski theorem or variants of it based on order-theoretic arguments. Such theorems do not provide conditions under which one has uniqueness of fixed points, and indeed fixed points need not be unique in general. Nevertheless, this question of uniqueness is interesting because it is closely related to coincidence of various standard models of programs as shown in [8], and this point is also discussed in Section 2. In [6,12], the issue of uniqueness was taken up and solved in the case of the operator T_P for the class of locally hierarchical programs, see [2], by methods entirely different from those employed in [2]. In fact, it was done by showing that Scott-Ershov domains, familiar in programming language semantics, can be turned into generalized ultrametric spaces in the sense of [10] and by then applying a fixed-point theorem to be found in [10]. In this paper, our main objective is to show how the approach of [6,12] can be extended to the operators Ψ_P and Φ_P in the context of the logic $\mathcal{FOUR}$. Indeed, our approach extends very generally to any many-valued logic whose associated space of valuations forms a domain under the construction we give later, see Theorem 3.4. We will confine our attention here to Ψ_P but we obtain, as a corollary, the fact that our results apply to Φ_P also and to T_P (trivially) in view of our earlier remarks.

---

* Department of Mathematics, University College Cork, Ireland, E-mail: e.clifford@student.ucc.ie , aks@ucc.ie

[1] In fact, we are working with a slight variant of Belnap's logic, due to Melvin Fitting.

Both authors wish to extend their thanks to the conference organizers, and to the Slovak University of Technology for the hospitality and support they received in presenting the results of this paper. They also wish to thank an anonymous referee for making several suggestions which helped to improve the presentation of the paper. The first-named author also wishes to thank the School of Mathematics, Applied Mathematics and Statistics at UCC, and the UCC Mathematical Society for their support.

ISSN 1335-3632

2 $\mathcal{FOUR}$–VALUED INTERPRETATIONS

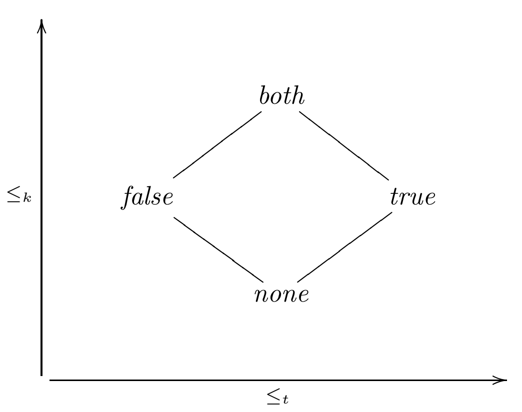
The logic $\mathcal{FOUR}$ has the four truth values *true (t)*, *false (f)*, *none (n)*, and *both (b)*. The first two of these are the familiar truth values of two-valued logic. The third truth value *none* (or *underdefined*) is found in Kleene's strong (and weak) three-valued logic (as *undefined*), and is the truth value given to something about which we have no information; it is also used in computation to represent non-termination. The fourth truth value *both* (or *overdefined*) can be thought of as the truth value given to something which we have been told is both *true* and *false*. Belnap in [1] offers some interesting motivation for this logic. He sees it as a means of dealing with a situation where a computer is relying on two different human operators, which may contradict each other. Fitting in [4] argues that it is an appropriate logic for handling conflicting information in distributed computing systems. In [14], Visser shows how this logic can be used as a means of investigating paradoxes such as that of the Liar.

Following [4], we note that $\neg t = f$; $\neg f = t$, $\neg n = n$ and $\neg b = b$. Furthermore, we define the operations $\wedge$ and $\vee$ by means of the following truth tables:

$\wedge$	n	f	t	b
n	n	f	n	f
f	f	f	f	f
t	n	f	t	b
b	f	f	b	b

$\vee$	n	f	t	b
n	n	n	t	t
f	n	f	t	b
t	t	t	t	t
b	t	b	t	b

We impose ordered structures on $\mathcal{FOUR}$ by defining the truth ordering $\leq_t$ and the knowledge ordering $\leq_k$ as in the Hasse diagram below.



It is an interesting and important fact that $\mathcal{FOUR}$ is a complete lattice in each of these orderings, and hence is a complete bilattice, see [4]. In the truth ordering, the bottom element is *false* and the top element is *true*. In the knowledge ordering, the bottom element is *none* and the top element is *both*. We note that negation is the left-right inversion of the Hasse diagram above. Further details of this are to be found in [4].

These orderings, and indeed any partial order on $\mathcal{FOUR}$, immediately extend to the set $I(X)$ of all functions I from any set X into $\mathcal{FOUR}$ when ordered pointwise by: $I_1 \leq I_2$ iff $I_1(A) \leq I_2(A)$ for all $A \in X$, where $\leq$ denotes either of the orderings $\leq_k$ and $\leq_t$, and we note that our usage of the symbol $\leq$ to order both truth values and functions should not cause any confusion. Indeed, the bottom resp. top element of $I(X)$ is simply the function identically equal to the bottom resp. top element of $\mathcal{FOUR}$ in the ordering $\leq$ on $\mathcal{FOUR}$. Moreover, given any family $M = \{I_k; k \in K\}$ of elements of $I(X)$, whether directed or not, the supremum $\sqcup M$ exists and is given by $\sqcup M(A) = \sqcup_{k\in K} I_k(A)$ for all $A \in X$. Similarly, the infimum $\sqcap M$ of M is given by $\sqcap M(A) = \sqcap_{k\in K} I_k(A)$ for all $A \in X$.

Now let P denote a normal logic program whose underlying first order language is $\mathcal{L}$; we refer to [9] for notation and basic facts concerning logic programming. Thus, P consists of a finite set of clauses of the form $A \leftarrow L_1, \dots, L_n$, where A is an atomic formula, called the *head* of the clause, and $L_1, \dots, L_n$ denotes a conjunction of literals L_i (atoms or negated atoms) called the *body* of the clause. In other words, a typical clause in P is of the form $A \leftarrow A_1, \dots, A_{n_1}, \neg B_1, \dots, \neg B_{n_2}$. It will be convenient to add two atoms *false* and *true* to $\mathcal{L}$. We let B_P denote the Herbrand base of P, that is, the set of all ground (or variable free) atoms formed from the symbols in $\mathcal{L}$. Taking $X = B_P$ in the previous paragraph, the set $I(B_P)$ is precisely the set of all $\mathcal{FOUR}$-valued valuations or interpretations in the usual sense of mathematical logic, where we always assume that $I(\mathit{false}) = \mathit{false}$ and $I(\mathit{true}) = \mathit{true}$ for any interpretation I. In future, we will denote the set $I(B_P)$ by $I_{P,4}$ and we note that it is a complete bilattice under the operations defined earlier.

In order to define the operators we want, we need first to define two sets P^* and P^{**} associated with P. To define P^*: first, put in P^* all ground instances of members of P; second, if a clause $A \leftarrow$ with empty body occurs in P^*, replace it with $A \leftarrow \mathit{true}$; finally, if the ground atom A is not yet the head of any member of P^*, add $A \leftarrow \mathit{false}$ to P^*. To define P^{**}: first, in P^* replace each ground clause $A \leftarrow L_1, \dots, L_n$ with $A \leftarrow L_1 \wedge \dots \wedge L_n$. Next, if there are several clauses in the resulting set having the same head, $A \leftarrow C_1$, $A \leftarrow C_2$, $\dots$, replace them with $A \leftarrow C_1 \vee C_2 \vee \dots$. Since there could be infinitely many members in P^* with the same head, we may have a countable disjunction at this point, but this is semantically unproblematic. We note that each ground atom A is the *head* of exactly one element $A \leftarrow C_1 \vee C_2 \vee \dots$ of P^{**}.

Following [4], we are now in a position to define the operator Ψ_P.

Definition 2.1. Let P be a normal logic program. We define the operator $\Psi_P \colon I_{P,4} \to I_{P,4}$ as follows. For any $I \in I_{P,4}$ and $A \in B_P$, we set

$$\Psi_P(I)(A) = I(C_1 \vee C_2 \vee \dots),$$

where $A \leftarrow C_1 \vee C_2 \vee \dots$ is the unique element of P^{**} whose head is A.

If we restrict attention to the truth values *true*, *false* and *none*, we obtain the conventional three-valued operator Φ_P, see [3,4], from this definition. If we further restrict to the truth values *true* and *false*, we obtain the two-valued operator T_P. In fact, as noted in [4], the form of the definition of Ψ_P just given suggests great generalization of such operators to any logic on any set of interpretations, even to the context of uncertain reasoning systems. However, this point of view combined with that of the current paper will be considered elsewhere. Indeed, by considering different definitions of conjunction and disjunction, it was shown in [5] how one may characterize different classes of programs by means of the operator Φ_P.

Next, we note that Ψ_P is monotonic for all programs P with respect to the $\leq_k$ ordering, and this important fact led Fitting [3] to his well-known treatment of negation using Φ_P. Moreover, Ψ_P is also monotonic with respect to the $\leq_t$ ordering for definite programs. Thus, using the Knaster-Tarski theorem, one always obtains least (v_k) and greatest (V_k) fixed points of Ψ_P relative to $\leq_k$, and, for definite programs, least (v_t) and greatest (V_t) fixed points of Ψ_p (and hence of T_P) relative to $\leq_t$. All these fixed points are different, in general. However, v_k, v_t and V_t are closely related, see [4, Proposition 15]. Indeed, as shown in [4], there are great advantages obtained by working in the bilattice $\mathcal{FOUR}$. Not only does one have a unified framework in which to study the two standard approaches to negation, but also, amongst other things, the interconnections between the fixed points can be stated in simple and elegant (algebraic) fashion. Furthermore, v_t and V_t play a fundamental role in logic programming semantics: the former being the least Herbrand model of a definite program; the latter, also a model of P, being fundamental in treatments of the completeness of SLDNF-resolution, see [2,9]. Of course, if Ψ_P has a unique fixed point, then all these fixed points coincide (with the well-known Clark-completion semantics) and this fact simplifies much of the analysis. There is another reason, also, why this situation is important, as follows.

One issue which is not addressed when applying the Knaster-Tarski theorem, and indeed cannot be, is the uniqueness (or otherwise) of the fixed points it provides (the Knaster-Tarski theorem says nothing about uniqueness). As already noted, classes of programs P for which T_P and Φ_P have unique respectively unique total fixed points are interesting. Indeed, using the operator Φ_P (and a variant of it) such classes were defined in [5,7,8] in a quite natural way. These classes were shown in [5,7,8] not only to be computationally adequate (can compute all partial recursive functions), but to be semantically unambiguous as well in that for each program in them the stable, well-founded, and weakly perfect models all coincide. They therefore provide an interesting framework within which to do logic programming, since one has simultaneously available within them both full computational power and a well-defined semantics which is the same in a number of the current, fashionable ways of viewing nonmonotonic reasoning. Thus, fixed-point theorems which supply uniqueness criteria have an important role in logic programming. One such is the theorem of Priess-Crampe and Ribenboim [10] which has found application in [10] to the operator T_P in discussing some specific examples considered in [11], see also [6] for applications of the multivalued version to disjunctive databases. Our intention here is to show how the Priess-Crampe and Ribenboim theorem can be applied, in conjunction with elementary domain theory, to the operator Ψ_P, and hence to Φ_P, for certain programs, and we proceed to do this next.

3 $I_{P,4}$ AS A DOMAIN AND AS AN ULTRAMETRIC SPACE

Let $(D, \sqsubseteq)$ denote a partially ordered set, or poset.

Definition 3.1. (1) A subset M of D is said to be *directed* if every finite subset of M has an upper bound in M (equivalently, if every pair of elements of M has an upper bound in M).
(2) We call $(D, \sqsubseteq)$ a *complete partial order (cpo)* if it has a bottom element $\bot$ and the supremum $\sqcup M$ of M exists in D for all directed subsets M of D.
(3) An element $x \in D$ is called *compact* iff whenever M is a directed subset of D and $x \sqsubseteq \sqcup M$, there exists $y \in M$ such that $x \sqsubseteq y$. We denote by D_C the set of compact elements of D.
(4) A subset A of D is called *consistent* if there exists $x \in D$ such that $a \sqsubseteq x$ for all $a \in A$. In particular, the set $\{a, b\} \subseteq D$ is *consistent* if there exists $x \in D$ such that $a \sqsubseteq x$ and $b \sqsubseteq x$.

Definition 3.2. Let $(D, \sqsubseteq)$ be a poset, and let D_C denote its set of compact elements. Then $(D, \sqsubseteq)$ is called a *Scott-Ershov domain* or simply a *domain*, see [13], if the following conditions hold:
(1) $(D, \sqsubseteq)$ is a cpo.
(2) For each $x \in D$, the set $\text{approx}(x) = \{a \in D_C; a \sqsubseteq x\}$ is directed and $x = \sqcup\,\text{approx}(x)$ (called the *algebraicity* of D).
(3) If $A \subseteq D$ is consistent, then $\sqcup A$ exists in D (called the *consistent completeness* of D).

Definition 3.3. Let $\leq$ denote a partial order on $\mathcal{FOUR}$ in which $\mathcal{FOUR}$ is a complete lattice with bottom element $\bot$. Then $I \in I_{P,4}$ is called *finite* if the set $\{A \in B_P; I(A) \neq \bot\}$ is finite. In particular, we define the finite interpretation $I_\bot$ by $I_\bot(A) = \bot$ for all $A \in B_P$.

Theorem 3.4. *Let $\leq$ denote a partial order on $\mathcal{FOUR}$ in which $\mathcal{FOUR}$ is a complete lattice with bottom element $\bot$. Then $(I_{P,4}, \leq)$ is a domain whose bottom element is $I_\bot$ and whose compact elements are the finite interpretations.*

Proof. First, because $(I_{P,4}, \leq)$ is a complete lattice, it is immediate that it is a cpo with bottom element $I_\bot$ and also that it is consistently complete.

Next, we show that any finite interpretation is a compact element. Suppose that I is a finite interpretation and let $I^\star = \{A \in B_P; I(A) \neq \bot\}$. Then $I^\star$ is a finite set, $I^\star = \{A_1, \dots, A_n\}$, say. Suppose $M = \{I_k; k \in K\}$ is a directed subset of $I_{P,4}$ such that $I \leq \sqcup M$. Thus, $I(A) \leq \sqcup_{k \in K} I_k(A)$ for all $A \in B_P$. Then, using the directedness of M, there is, for each $i = 1, \dots, n$, $I_{k_i} \in M$

such that $I(A_i) \leq I_{k_i}(A_i)$. Since $\{I_{k_i}; i = 1, \dots, n\}$ is finite, and using again the fact that M is directed, there exists $J \in M$ such that $I_{k_i} \leq J$ for $i = 1, \dots, n$. But then $I \leq J$ as required and therefore I is a compact element of $I_{P,4}$.

Conversely, we show that the compact elements of $(I_{P,4}, \leq)$ are the finite interpretations. Let M be the set of all finite interpretations. Then M is directed. To see this, let $I_1, I_2 \in M$. Define I_3 by $I_3(A) = \sqcup\{I_1(A), I_2(A)\}$ for all $A \in B_P$. Then $I_1 \leq I_3$ and $I_2 \leq I_3$ and clearly I_3 is a finite interpretation. Thus, $I_3 \in M$ also. Hence, M is a directed subset of $I_{P,4}$. Now suppose that I is a compact element of $I_{P,4}$. Then trivially we have $I \leq \sqcup M$, since $\sqcup M$ is the interpretation whose value on all elements of B_P is equal to the top element of $\mathcal{F}OUR$ in the given ordering on $\mathcal{F}OUR$. Thus, by directedness of M and the compactness of I, there exists $J \in M$ such that $I \leq J$. Since J is a finite interpretation, it follows that I is finite also. Therefore, the compact elements of $(I_{P,4}, \leq)$ are finite interpretations and indeed we now see that the compact elements of $(I_{P,4}, \leq)$ are precisely the finite interpretations.

We show next that for any $I \in I_{P,4}$, approx(I) is directed. Let $I_1, I_2 \in$ approx(I). Then I_1 and I_2 are finite interpretations with $I_1 \leq I$ and $I_2 \leq I$. Again, define I_3 by $I_3(A) = \sqcup\{I_1(A), I_2(A)\}$ for all $A \in B_P$. Then by definition of supremum, we have $I_1 \leq I_3 \leq I$ and $I_2 \leq I_3 \leq I$, and of course I_3 is finite. Thus, $I_3 \in$ approx(I), and so approx(I) is directed. Therefore, approx(I) is directed for any $I \in I_{P,4}$.

Finally, we show that for any $I \in I_{P,4}$, we have $I = \sqcup$ approx(I). Clearly, by definition of approx(I) and of supremum, we have that $\sqcup$ approx$(I) \leq I$. Let $A \in B_P$. Define $I_A \in I_{P,4}$ by $I_A(A) = I(A)$, and $I_A(B) = \perp$ for all $B \neq A$. Then clearly $I_A \in$ approx(I). Also, for all $A \in B_P$ we have $I(A) = I_A(A) \leq \sqcup$approx$(I)(A)$. Thus, $I \leq \sqcup$ approx(I) and it follows that $I = \sqcup$ approx(I), as required, and the proof is complete.

Of course we obtain, as corollaries of this result, that $I_{P,4}$ is a domain in both of the two orderings we have been considering on $\mathcal{F}OUR$.

We now turn our attention to generalized ultrametric spaces.

Definition 3.5. 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*) or simply an *ultrametic space* if $d\colon X \times X \to \Gamma$ 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 $d(x, z) \leq \gamma$.

Definition 3.6. 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 simply a *ball* in X with centre x and radius γ.

Definition 3.7. An ultrametric space X is called *spherically complete* if $\cap\mathcal{C} \neq \emptyset$ for any chain $\mathcal{C}$ of balls in X (a *chain of balls* is a set of balls which is totally ordered by inclusion).

This brings us to an important theorem of Priess-Crampe and Ribenboim, see [10], which we state in a reduced form sufficient for our present purposes.

Theorem 3.8. *Let (X, d) be a spherically complete ultrametric space and let $f\colon X \to X$ be strictly contracting in the sense that $d(f(x), f(y)) < d(x, y)$ for all $x, y \in X$ with $x \neq y$. Then f has a unique fixed-point.*

It is our intention to apply this theorem to Ψ_P. To do this, we first give a general construction which turns a domain, and $I_{P,4}$ in particular, into a generalized ultrametric space.

Let γ denote an arbitrary countable ordinal, and let Γ_γ denote the set $\{2^{-\alpha}; \alpha \leq \gamma\}$ of symbols $2^{-\alpha}$ ordered by $2^{-\alpha} < 2^{-\beta}$ if and only if $\beta < \alpha$, and denote $2^{-\gamma}$ by 0. Thus, Γ_γ is essentially $\gamma + 1$ endowed with the reverse order, but for historical reasons we prefer to work with the set Γ_γ, see [6]. Now let $(D, \sqsubseteq)$ be a domain, with set D_C of compact elements.

Definition 3.9. Let $r\colon D_C \to \gamma$ be a function, called a *rank function*, and form Γ_γ. We define the distance function $d_r\colon D \times D \to \Gamma_\gamma$ by $d_r(x, y) = \inf\{2^{-\alpha};$ for every $c \in D_C$ with $r(c) < \alpha$ we have $c \sqsubseteq x$ if and only if $c \sqsubseteq y\}$.

It turns out that d_r is an ultrametric which is said to be *induced by* r, see [6,12]. In fact, the following theorem was established in [6,12].

Theorem 3.10. *The ultrametric space (D, d_r) is spherically complete.*

4 UNIQUE FIXED POINTS OF Ψ_P

Suppose now that P is a normal logic program. A *level mapping* for P is a mapping $l\colon B_P \to \gamma$, where γ is a countable ordinal (not necessarily the first infinite ordinal). Fix an ordering $\leq$, such as $\leq_k$ or $\leq_t$, in which $\mathcal{F}OUR$ is a complete lattice with bottom element $\perp$; then $I_{P,4}$ is also a complete lattice. By Theorem 3.4, $I_{P,4}$ is a domain whose compact elements are the finite interpretations. Define the rank function r_l induced by l as follows: we put $r_l(I_\perp) = 0$ and, for every finite interpretation $I \neq I_\perp$, we set $r_l(I) = \max\{l(A); A \in B_P$ and $I(A) \neq \perp\}$. We denote by d_l the ultrametric resulting from r_l in accordance with Definition 3.9. Indeed, it is easy to see that d_l has a simpler, equivalent definition, as follows: if $I_1 = I_2$, then $d_l(I_1, I_2) = 0$; otherwise $d_l(I_1, I_2) = 2^{-\alpha}$, where I_1 and I_2 differ (i.e. $I_1(B) \neq I_2(B)$) on some ground atom B with $l(B) = \alpha \leq \gamma$ but agree (i.e. $I_1(A) = I_2(A)$) on all ground atoms A of lower level.

Level mappings have proved to be important in logic programming in a number of contexts including studies concerned with termination and completeness. One of their main uses is the provision of syntactic conditions which identify tractable classes of programs by prohibiting "negation through recursion", that is, by preventing an atom occurring in the head of a clause and simultaneously occurring negated in its body. This is illustrated by the following definition. Suppose $A \leftarrow A_1, \dots, A_{n_1}, \neg B_1, \dots, \neg B_{n_2}$ is a typical ground instance of a clause in P, where $n_1, n_2 \geq 0$. We call P *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 for each clause, and we call P *locally hierarchical* (with respect to l) if the inequalities $l(A) > l(A_i), l(B_j)$ hold for all i and j for each clause. Both of the classes defined here have turned out to be important in logic programming.

Our main theorem is the following result which is an extension to Ψ_P of an earlier result established in [2,6,12] for T_P.

Theorem 4.1. *Let P be a normal logic program which is locally hierarchical with respect to a level mapping l. Then Ψ_P is strictly contracting with respect to d_l and hence has a unique fixed point.*

P r o o f . Let $I_1, I_2 \in I_{P,4}$ be such that $d_l(I_1, I_2) = 2^{-\alpha}$. There are two cases to consider.

Case 1: $\alpha = 0$. In this case, I_1 and I_2 differ on some ground atom of level 0. Let $A \in B_P$ be arbitrary with $l(A) = 0$. Consider $\Psi_P(I_1)$ and $\Psi_P(I_2)$. By the hypothesis on P and the fact that $l(A) = 0$, the element $A \leftarrow C_1 \vee C_2 \dots$ in P^{**} with A in its head must either be of the form $A \leftarrow \mathit{true}$ or $A \leftarrow \mathit{false}$. But $I_1(\mathit{true}) = I_2(\mathit{true}) = t$ and $I_1(\mathit{false}) = I_2(\mathit{false}) = f$. Thus, we either have $\Psi_P(I_1)(A) = I_1(\mathit{true}) = I_2(\mathit{true}) = \Psi_P(I_2)(A)$ or we have $\Psi_P(I_1)(A) = I_1(\mathit{false}) = I_2(\mathit{false}) = \Psi_P(I_2)(A)$. Hence, $\Psi_P(I_1)$ and $\Psi_P(I_2)$ agree on all ground atoms of level 0, and it follows that $d_l(\Psi_P(I_1), \Psi_P(I_2)) < 2^{-0} = d_l(I_1, I_2)$.

Case 2: $\alpha > 0$. In this case, I_1 and I_2 differ on some ground atom of level α, but agree on all ground atoms of lower level. Let $A \in B_P$ with $l(A) \leq \alpha$. Consider the unique element $A \leftarrow C_1 \vee C_2 \vee \dots$ in P^{**} with A as its head. Since P is locally hierarchical, each atom occurring in each clause body C_i has level strictly less than α. Therefore, $I_1(C_1 \vee C_2 \vee \dots) = I_2(C_1 \vee C_2 \vee \dots)$, by our hypothesis. Hence, $\Psi_P(I_1)(A) = I_1(C_1 \vee C_2 \vee \dots) = I_2(C_1 \vee C_2 \vee \dots) = \Psi_P(I_2)(A)$. Thus, $\Psi_P(I_1)$ and $\Psi_P(I_2)$ agree on all ground atoms of level $\leq \alpha$. Hence, $d_l(\Psi_P(I_1), \Psi_P(I_2)) < 2^{-\alpha} = d_l(I_1, I_2)$.

Since cases 1 and 2 cover all possiblities, we see that Ψ_P is strictly contracting with respect to d_l. Finally, $(I_{P,4}, d_l)$ is spherically complete by Theorem 3.10, and thus the required second conclusion follows from Theorem 3.8.

It follows from our earlier remarks that under the hypothesis of the previous theorem, both T_P and Φ_P are strictly contracting and hence also have unique fixed points. Indeed, the first of these comments was established in [6,12], as already observed, and it was this fact that led to the present extension to Φ_P and Ψ_P. Finally, we note that certain of these ideas have been generalized in another direction in [7].

References

[1] BELNAP, Jr., N. D.: A Useful Four-Valued Logic, In: Modern Uses of Multiple-Valued Logic, Dunn, J.M. and Epstein, G. (Eds.), D. Reidel, 1977, pp. 5–37.

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

[3] FITTING, M. C.: A Kripke-Kleene Semantics for Logic Programming, Journal of Logic Programming **2** (1985), 295-312.

[4] FITTING, M. C.: Fixpoint Semantics for Logic Programming: A Survey, Theoretical Computer Science, 32 pages, to appear.

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

[6] HITZLER, P.—SEDA, A. K.: The Fixed-Point Theorems of Priess-Crampe and Ribenboim in Logic Programming, Proceedings of the International Conference and Workshop on Valuation Theory, University of Saskatchewan in Saskatoon, Canada, July 1999. Fields Institute Communications Series, American Mathematical Society, Providence, R.I., 17 pages, to appear.

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

[8] HITZLER, P.—SEDA, A. K.: Localizing Uniquely Determined Programs, Preprint, Department of Mathematics, University College, Cork, Ireland, pp. 1–16. Submitted to the Proceedings of MFCSIT 2000, July, 2000.

[9] LLOYD, J. W.: Foundations of Logic Programming, Springer, Berlin, 1988.

[10] PRIESS-CRAMPE, S.—RIBENBOIM, P.: Ultrametric Spaces and Logic Programming, Journal of Logic Programming **42** (2000), 59–70.

[11] SEDA, A. K.: Quasi-metrics and the Semantics of Logic Programs, Fundamenta Informaticae **29**(2) (1997), 97–117.

[12] SEDA, A. K.—HITZLER, P.: Topology and Iterates in Computational Logic, Proceedings of the 12th Summer Conference on General Topology and its Applications: Special Session on Topology in Computer Science, Ontario, August 1997, Topology Proceedings **22** (1999), 427–469.

[13] STOLTENBERG-HANSEN, V.—LINDSTRÖM, I.—GRIFFOR, E. R.: Mathematical Theory of Domains, Cambridge Tracts in Theoretical Computer Science No. 22, Cambridge University Press, Cambridge, 1994.

[14] VISSER, A.: Four-Valued Semantics and the Liar, Journal of Philosophical Logic **13** (1984), 181–212.

Received 14 June 2000

Eleanor Clifford (BA) is a postgraduate student in mathematics at University College Cork, Ireland working under the direction of A.K. Seda. Her interests include logic, philosophy and their connections with mathematics.

Anthony Karel Seda (Dr., C.Math, FIMA) is a Senior Lecturer in the Department of Mathematics, University College Cork. His research interests in the past were concerned with mathematical analysis, but now are concerned with the mathematical foundations of computer science and, in particular, of computational logic and of artificial intelligence.