Topology and Logic Programming Semantics

Diplomarbeit
im Fach Mathematik

vorgelegt von
Pascal Hitzler

im Wintersemester 1997/98
an der Eberhard-Karls-Universität Tübingen
unter der Betreuung von Prof. Dr. H. Salzmann.

Topology and Logic Programming Semantics

Diplomarbeit
im Fach Mathematik

vorgelegt von
Pascal Hitzler

im Wintersemester 1997/98
an der Eberhard-Karls-Universität Tübingen
unter der Betreuung von Prof. Dr. H. Salzmann.

Preface

Logic programming employs logic as a programming language. Thus a logic program consists of a set of clauses of a certain form, most often a subset of the clauses of first order logic, viewed as axioms. Computation in this paradigm is deduction from these axioms via some interpreter.

Logic programming semantics is concerned with background theory for logic programming. It tries to provide models for logic programs to give them their "intended meaning" and to connect them with practically implementable interpreters.

In this thesis, we are concerned with the semantics of normal logic programs, as intoduced in Section 1. Prolog, for example, surely the best known interpreter for logic programs, is basically designed to deal with normal logic programs (though there are usually some additional features implemented). The discussion of interpreters and their connection with the models we obtain will only be of minor importance for us, and so we focus on finding suitable models for normal logic programs.

The main tool used in logic programming semantics is the immediate consequence operator, introduced in Definition 1.1, which can be derived from any normal logic program. Its importance comes from the fact that the models for some given normal logic program are exactly the pre-fixed points of the immediate consequence operator. As will be seen in Section 1, models which are also fixed points of this operator are of even greater importance.

So, roughly speaking, our task is to find fixed points of the immediate consequence operator. Since some of the most common fixed point theorems, such as the Banach Contraction Mapping Theorem 4.1 or the Knaster-Tarski Theorem 2.3, are basically topological in nature, we are immediately in the realm of set-theoretic topology, and this thesis is concerned with displaying how topology can be employed in the area of logic programming semantics.

We suppose that the reader is familiar with basic concepts from topology and logic as given, for example, in [Wil70] and [EFT94]. We will shortly review the most important notions from logic programming which are used in the sequel in Section 1, basically showing the importance of the immediate consequence operator. Some familiarity with logic programming is presupposed, and our main reference to this area is [Llo88].

Section 2 is devoted to the classical theory of normal logic programs without negation, called definite logic programs. In Section 2.1, we introduce the Scott topology on domains and discuss some of their properties, including the Knaster-Tarski Theorem. In Section 2.2, we apply the results to definite logic programs and settle the issue of semantics for these. Additionally, we characterize the Scott topology on the space I_P of all Herbrand interpretations of a given logic program P, usually given by pure order-theoretic notions, by using logical notions only. Our main reference for the first part is [SLG94] and for the second part [Llo88] and [Sed95].

In Section 3, we define the atomic topology Q on I_P, which will be shown to coincide with the Cantor topology on I_P. Topological properties of Q are discussed in Section 3.1, whereas in Section 3.2, it will be seen that there is a strong connection between models of a normal logic program P and convergence in Q, and between supported models and continuity of the immediate consequence operator in Q. Our main reference for this part is [Sed95], although Theorems 3.6 and 3.8 are new.

In Section 4, we will use the Banach Contraction Mapping Theorem and a closely related theorem by Priess-Crampe and Ribenboim to derive fixed points of the immediate consequence operator. In Section 4.1, we will provide an extended ultrametric on every domain, hence on I_P, and in Section 4.2, we will settle the semantics of strictly level-decreasing programs, as defined there. The main references are [PR97] and for the new results [SH97c].

Since, for arbitrary normal logic programs, the immediate consequence operator in general has more than one fixed point, the Banach Contraction Mapping Theorem is only of limited use. We therefore employ quasi-metric spaces in Section 5. The Rutten-Smyth Theorem 5.5 generalizes both the Knaster-Tarski Theorem and the Banach Contraction Mapping Theorem and is used in Section 5.1 to recover the classical theory from Section 2. In Section 5.2, we will use level mappings to define a quasi-metric on I_P, and show that this quasi-metric is strongly connected with the atomic topology Q. Our main reference is [Rut96] for Theorem 5.5 and [Sed97] for the results on logic programs.

Section 6 deals with compactness properties of generalized metric spaces, and is only indirectly connected with Section 5, providing some background theory. Some well-known results from metric spaces are generalized. In Section 6.1, it is shown that total boundedness and sequential completeness imply sequential compactness and, in Section 6.2, that total boundedness and net completeness imply compactness. It should be noted here that quasi-metric spaces are, in general, not second countable. The results of this section are new.

In Section 7, we use a slightly different approach. It will be seen that it is useful to partition a given logic program into subprograms and to apply the immediate consequence operators of the subprograms subsequently. Section 7.1 deals with stratified logic programs and a new characterization of their standard model is given, as well as an alternative way of finding models for a subclass of these. A subclass of the class of all locally stratified programs is examined in section 7.2, and a new way of obtaining their standard model is presented. Our main reference for the first part is [ABW88], and [SH97c] for the new results and the second part.

A comparison of the different approaches employed in the sequel is made in Section 8.

All the example programs studied throughout the thesis have been moved to Appendix A, so that the different approaches can easily be compared. Some additonal interesting programs are also provided.

At the end of each section, some open problems which are raised by the results are given, and they are collected in Appendix B.

The present thesis was written during a stay at University College Cork, Ireland, while working with A.K. Seda on [SH97c]. Thus, most of the new results can be found there as indicated above. The problems leading to these results were suggested by A.K. Seda and worked out under his guidance during the above mentioned stay at University College Cork.

A large number of people were important for me during the course of my studies, and I cannot possibly list everyone whom I would like to thank for his or her support, be it in personal or professional matters. Everyone of them made my life richer and I love to remember the moments they gave me.

I would like to thank Prof. Dr. H. Salzmann for his support. He let me have my own way for writing this thesis and especially made it possible for me to stay at University College Cork for this purpose.

I would like to thank Dr. A.K. Seda for all his help and the numerous discussions about mathematical and non-mathematical matters. He opened the door for me to the mathematical area to which this thesis belongs.

I would like to thank Prof. Dr. G. Kalmbach for all the work I could and still can do with her, for her support, and for all the talks about academic life that gave me insights I could otherwise not possibly have at this stage.

I would like to thank PD Dr. R. Bödi for his always open door and for his readiness to answer all the questions I asked him.

I would like to thank Mike and Ralf. Their influence on my life is probably higher than they can imagine.

I would like to thank Almut and Andi for all the emotional support they gave me during the last three years. My life would not be as rich without them.

I would like to thank Anne for her love. There is only one place I can call my home.

I would like to thank my family, and first of all my mother. It is good to know a place where you can go when everything else fails.

Contents

Preface i

1 Introduction 1

2 Definite Programs 5
2.1 The Scott Topology on Domains 5
2.2 Application to Definite Logic Programs 8

3 The Atomic Topology 11
3.1 The Atomic Topology Q 11
3.2 Models and the Atomic Topology 12

4 Metric and Extended Metric Spaces 17
4.1 Domains as Extended Ultrametric Spaces 17
4.2 Application to Strictly Level-decreasing Programs 20

5 Quasi-metric Spaces 25
5.1 Domains as Quasi-ultrametric Spaces 25
5.2 Quasi-metrics and the Atomic Topology 28

6 Compactness of Generalized Metric Spaces 31
6.1 Sequential Compactness 31
6.2 Compactness 32

7 Partitioning Programs 35
7.1 Stratified Programs 35
7.2 Semi-strictly Level-decreasing Progams 39

8 Conclusions 45

Appendix 47

A Programs 47

B Problems 51

Literature 53

1 Introduction

We shortly review the most important concepts from the theory of logic programming needed in this paper, focussing on the importance of the immediate consequence operator.

A (*normal*) *logic program* consists of a finite set of *program clauses*, for convenience simply called *clauses*, which are clauses from first order logic of the form

$$\forall(A \leftarrow L_1 \wedge \cdots \wedge L_n), \text{ written as } A \leftarrow L_1, \ldots, L_n,$$

where $n \in \mathbb{N}$ is allowed to be zero (such a clause is called a *unit clause* or simply a *fact*), and L_i $(i = 1, 2, \ldots, n)$ are literals, that is, positive or negative atoms within some given first order language, and A an atom in this language. A logic program is called *definite* if all literals in every clause of the program are positive. Thus, a logic program is definite if no negation symbol occurs in it. In the above clause, A is called the *head* of the clause and $L_1, \ldots, L_n$ is called the *body* of the clause, where the commas stand for conjunction. Consequently, each L_i is called a *body literal* of the clause.

Given a logic program P, there is always a (minimal) underlying first order language, denoted by $\mathcal{L}_P$, whose constant, function, variable, and predicate symbols are the constant, function, variable, and predicate symbols occuring in P, respectively. If P does not contain a constant symbol, we add one, so that $\mathcal{L}_P$ always contains at least one constant symbol. The *Herbrand universe* U_P of P is the set of all *ground terms* over $\mathcal{L}_P$, i.e. all terms obtained from the constant and function symbols in $\mathcal{L}_P$. The *Herbrand base* B_P of P is the set of all *ground* atoms over $\mathcal{L}_P$, i.e. the set of all expressions obtained by taking (ground) terms from U_P as arguments for predicate symbols in $\mathcal{L}_P$. Given a clause C in P, a *ground instance* of C is obtained by assigning an element of U_P to every variable symbol occuring in C.

The *Herbrand preinterpretation* for P assigns all constant and function symbols to themselves, thus taking U_P as the domain of the preinterpretation. Every subset I of B_P can thus be identified with a *Herbrand interpretation* in the following way. Given $I \subseteq B_P$, assign the truth value "true" to all ground atoms in I and "false" to all ground atoms not in I. Given an interpretation, let $I := \{A \in B_P \mid A \text{ evaluates to "true" in that interpretation}\}$. So the set $I_P := \{I \mid I \subseteq B_P\} = \{\chi \mid \chi : B_P \to \mathbf{2}\} = \mathbf{2}^{B_P} = \prod_{A \in B_P} \mathbf{2}$, where $\mathbf{2} = \{0, 1\}$, can be identified with the set of all intepretations for P. Obviously, I_P, ordered by set inclusion, is a complete lattice.

A *(Herbrand) model* for P is an interpretation $I \in I_P$ such that every clause in ground(P) evaluates to "true" with respect to I. Here, ground(P) denotes the set of all ground instances of clauses in P. A model M for P is called *minimal* if there is no model for P strictly contained in M. An interpretation (a model) I for P is called *supported* if, for every $A \in I$, there exists a clause $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ in ground(P) such that $A_k \in I$ and $B_l \notin I$ for $k = 1, \ldots, k_1$ and $l = 1, \ldots, l_1$. From a programmer's point of view, supported models are closer to the "intended meaning" of a program than models which are not supported.

We define the main tool employed in the sequel:

1.1 Definition For any given logic program P, the *single step operator* or *immediate consequence operator* is defined as a function $T_P : I_P \to I_P$ by

$$T_P(I) := \{A \in B_P \mid \text{ there is a clause } A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1} \\ \text{in ground}(P) \text{ with } A_k \in I \text{ and } B_l \notin I \text{ for } k = 1, \ldots, k_1, l = 1, \ldots, l_1\}.$$

The following observations connect the single step operator to (supported) models, and are therefore crucial to the theory of logic programming semantics.

1.2 Theorem (see [Llo88]) Let P be a logic program and let $M \in I_P$. Then M is a model for P if and only if $T_P(M) \subseteq M$.

Proof: Suppose M is a model for P and $A \in T_P(M)$. Then there is a clause $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ in ground(P) such that $A_k \in M$ and $B_l \notin M$ for $k = 1, \ldots, k_1$, $l = 1, \ldots, l_1$. Since the above clause evaluates to "true" with respect to M, $A \in M$.

Conversely, suppose $T_P(M) \subseteq M$ and let $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ be a clause in ground(P) with $A_k \in M$ and $B_l \notin M$ for $k = 1, \ldots, k_1$, $l = 1, \ldots, l_1$. Then by definition of T_P, $A \in T_P(M) \subseteq M$ as required. ∎

Thus models of P are exactly the *pre-fixed points* of the immediate consequence operator T_P. The following proposition shows that supported interpretations are exactly the *post-fixed points* of T_P.

1.3 Proposition (see [ABW88]) Let P be a logic program and $I \in I_P$. Then I is supported if and only if $I \subseteq T_P(I)$.

Proof: Suppose $I \in I_P$ is a supported interpretation of P and $A \in I$. Then there is a clause $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ in ground(P) such that $A_k \in I$ and $B_l \notin I$ for $k = 1, \ldots, k_1$, $l = 1, \ldots, l_1$, and therefore $A \in T_P(I)$.

Conversely, let $I \in I_P$ with $I \subseteq T_P(I)$ and let $A \in I$. Since $A \in T_P(I)$, there must be a clause in ground(P) with head A such that its body evaluates to "true" with respect to I. Therefore I is supported. ∎

Together, these results give us the following theorem.

1.4 Theorem (see [ABW88]) Let P be a logic program and let $M \in I_P$. Then M is a supported model for P if and only if M is a fixed point of T_P.

Our main tool for finding models for a given logic program employs sequences constructed by iterating the single step operator. For convenience, we define recursively for a given set X, point $x \in X$ and function $f : X \to X$,

$$f^0(x) = x \text{ and} \\ f^{n+1}(x) = f(f^n(x)) \text{ for all } n \geq 0.$$

Occasionally, we will refer to the sequence $(f^n(x))_{n\in\mathbb{N}}$ as the *orbit of x under f.*

Now suppose we have a topology $\mathcal{T}$ on I_P such that limits of sequences are unique, T_P is continuous with respect to $\mathcal{T}$, and the orbit $(T_P^n(I))$ of some $I \in I_P$ under T_P converges in $\mathcal{T}$ to some $M \in I_P$. Then M is a supported model for P since $T_P(M) = T_P(\lim T_P^n(I)) = \lim T_P(T_P^n(I)) = \lim T_P^{n+1}(I) = \lim T_P^n(I) = M$.

This observation will lead our thoughts throughout the sequel. Both the Knaster-Tarski Theorem studied in Section 2 and the Banach Contraction Mapping Theorem studied in Section 4 employ this or a similar principle. We will study a variety of different topologies or topology-like structures on I_P in order to obtain models for normal logic programs.

We introduce some notions used in the sequel.

Given a sequence $(x_n)_{n\in N}$ with some index set N, we often write this sequence as $(x_n)_n$, (x_n), or even x_n if it is clear from the context what the index set is. For a sequence (x_n), we say that a property holds *eventually* for (x_n) if there is a $k_0 \in \mathbb{N}$ such that the property holds for the sequence $(x_k)_{k\geq k_0}$. The notion of *limit* of a sequence (or a net) (I_λ), i.e. $\lim I_\lambda = I$ or $I_\lambda \to I$, always refers to the topology or the structure (in Section 5) currently discussed. If it refers to another structure, this is explicitly stated.

For a given set X, we denote the power set of X by $\mathbf{2}^X$; it can be identified with the set of all functions from X to $\mathbf{2}$, as usual.

We let ω denote the first infinite ordinal; we note that it can be identified with the set $\mathbb{N}$ of natural numbers (here always including 0). As usual, we set $\alpha + 1$ to be the successor of the ordinal α, $\alpha + 2$ as the successor of $\alpha + 1$ and so on, thus obtaining $\omega 2 = \bigcup_{n\in\mathbb{N}} \omega + n$. Iterating this, we obtain ωn, $\omega^2 = \bigcup_{n\in\mathbb{N}} \omega n$, ω^ω and so on. For convenience[1], we will denote the set of all ordinals obtained thus by Ω. Note that all of its members are countable ordinals. As is well known, every ordinal $\alpha_0 \in \Omega$ can be identified with the set $\{\alpha \mid \alpha < \alpha_0\}$ in the usual ordering.

Given any partially ordered set $(X, \leq)$ and a subset $A \subseteq X$, we denote the least upper bound and the greatest lower bound, when these exist, by $\sup A$ and $\inf A$, respectively. A *chain* in X is a subset $A \subset X$ such that, for all $a, b \in A$, either $a \leq b$ or $b \leq a$. An ω-*chain* in X is a sequence $(x_n)_{n\in\mathbb{N}}$ in X such that $x_n \leq x_{n+1}$ for every $n \in \mathbb{N}$.

We will use the notation $a := b$ or $b =: a$ for emphasizing that we define a to be b.

[1]Note, that this is not standard notation.

2 Definite Programs

We introduce domains and the Scott topology and discuss some of their features, including the Knaster-Tarski Theorem, in Section 2.1, and we apply these results to definite logic programs in 2.2. Finally, we give an alternative characterization of the Scott topology on I_P using logical notions only.

2.1 The Scott Topology on Domains

The Knaster-Tarski Theorem is a well-known fixed point theorem for partially ordered sets with some additional properties, as studied below. It is usually stated for Scott-domains or complete lattices. We give it in a much weaker form. The results in this section are taken from [SLG94].

2.1 Definition (ω-cpo) A partially ordered set $(D, \leq)$ is called an *ω-complete partial order* (*ω-cpo*) if

(1) there exists $\perp \in D$ such that for all $a \in D$ we have $\perp \leq a$ ($\perp$ is called the *bottom element* of D) and

(2) if $a_0 \leq a_1 \leq \dots$ is an ω-chain in D, then $\sup_{i \in \mathbb{N}} a_i$ exists in D.

2.2 Definition (ω-continuity) Let D and E be ω-cpos and $f : D \to E$ a function.

(1) f is called *monotonic* if $a \leq b$ implies $f(a) \leq f(b)$ for all $a, b \in D$.

(2) f is called *ω-continuous* if f is monotonic and for every ω-chain $a_0 \leq a_1 \leq \dots$ we have $f(\sup_{i \in \mathbb{N}} a_i) = \sup_{i \in \mathbb{N}} f(a_i)$

2.3 Theorem (Knaster-Tarski) Let D be an ω-cpo and let $f : D \to D$ be an ω-continuous function. Then f has a least fixed point a. Furthermore, $a = \sup_{n \in \mathbb{N}} f^n(\perp)$.

The general idea of the proof is to iterate f on $\perp$, obtaining the orbit $\perp$ under f, which is an ω-chain by monotonicity of f. Its supremum is the least fixed point of f, by using ω-continuity of f. Theorem 5.5 will generalize this result and a detailed proof will be given there.

The spaces under consideration when Theorem 2.3 is applied are usally stronger then ω-cpos. We continue with introducing these. Recall that a partially ordered set A is called *consistent* if it has an upper bound, and is called *directed* if every finite subset of A has an upper bound in A.

2.4 Definition (Scott-Ershov-domain) A partial ordered set $(D, \leq)$ is called a *complete partial order* (*cpo*) if

(1) there exists $\perp \in D$ such that for all $a \in D$ we have $\perp \leq a$ ($\perp$ is called the *bottom element* of D) and

(2) if $A \subseteq D$ is a directed set, then $\sup A$ exists in D.

An element c of a cpo is called *compact* (or *finite*) if, for every directed set $A \subseteq D$ with $c \leq \sup A$, there is some $a \in A$ with $c \leq a$. We denote the set of all compact elements of D by D_c.
A cpo D is called a (*Scott-Ershov-*) *domain* if

(1) for every $a \in D$ the set $\mathrm{approx}(a) := \{c \in D_c \mid c \leq a\}$ is directed, $a = \sup \mathrm{approx}(a)$ (D is *algebraic*) and

(2) for every $\{a, b\} \subseteq D_c$ which is consistent, $\sup\{a, b\}$ exists in D.

Note that every cpo is already an ω-cpo.

Intuitively, $x \leq y$ in a domain can be interpreteted as "x approximates y". Compact elements can be considered as practically implementable objects in a computer system, so that every object of interest can be arbitrarily closely approximated by those.

In Section 4, we will need the following proposition, which again can be found in [SLG94].

2.5 Proposition Every domain D is *consistently complete*, i.e. every consistent set in D has a supremum.

Proof: Note first, that if $\{a, b\} \subseteq D_c$ is consistent, then $\sup\{a, b\} \in D_c$. Indeed, let $A \subseteq D$ be directed with $\sup\{a, b\} \leq \sup A$. Then there are $x, y \in A$ such that $a \leq x$ and $b \leq y$. Since A is directed, there is $z \in A$ with $x, y \leq z$ and therefore $\sup\{a, b\} \leq z \in A$ as required. From this it is a straightforward proof by induction that, for every finite $C \subseteq D_c$ which is consistent, $\sup C$ exists in D_c.

Now let $A \subseteq D$ be consistent. If $A = \emptyset$, then $\sup A = \perp \in D$, so suppose $A \neq \emptyset$. Let x be an upper bound of A and $B := \bigcup_{y \in A} \mathrm{approx}(y) \subseteq D_c$. Then B is consistent with x as upper bound. Now $B_1 := \{\sup C \mid C \subseteq B, C \text{ finite}\} \subseteq D_c$ by the preceeding observation. Since $\sup\{\sup C_1, \sup C_2\} = \sup C_1 \cup C_2$, B_1 is directed and therefore $\sup B_1$ exists in D. It remains to show that $\sup B_1 = \sup A$. For $y \in A$, we have $\mathrm{approx}(y) \subseteq B_1$, hence $y = \sup \mathrm{approx}(y) \leq \sup B_1$, so $\sup B_1$ is an upper bound of A. Let z be another upper bound of A and $C \subseteq B_1$ be finite. Then z is an upper bound of B, and therefore of C. So we get $\sup C \leq z$ for every choice of C and hence $\sup B_1 \leq z$. ■

It follows immediate from the previous proposition that domains are exactly the consistently complete algebraic cpos.

We proceed with studying the appropriate notion of continuity for domains.

2.6 Definition (Scott-continuity) Let D, E be domains and $f : D \to E$ a function. We call f *Scott-continuous* if, for every directed set $A \subseteq D$, $f(A)$ is directed and $f(\sup A) = \sup f(A)$.

Note that every Scott-continuous function is ω-continuous. Indeed, if $x \leq y$ in D, then $A = \{x, y\}$ is directed and therefore $f(A) = \{f(x), f(y)\}$ is directed. Furthermore,

$f(y) = f(\sup A) = \sup f(A) = \sup\{f(x), f(y)\}$ and hence $f(x) \leq f(y)$, which shows monotonicity.

The topology underlying the notion of Scott-continuity is introduced next.

2.7 Definition (Scott topology) Let D be a domain. A set $O \subseteq D$ is open with respect to the *Scott topology $\mathcal{S}$ on D* if and only if

(1) if $a \in O$ and $a \leq b$ then $b \in O$ and

(2) if $a \in O$ then there exists $c \in \mathrm{approx}(a)$ with $c \in O$.

Note that the collection of all sets $\{\uparrow c \mid c \in D_c\}$ where $\uparrow c = \{a \in D \mid c \leq a\}$ is a basis for $\mathcal{S}$ on D. Thus, $\mathcal{S}$ is second countable whenever D_c is countable.

To show that continuity with respect to $\mathcal{S}$ is exactly Scott-continuity, the following characterization will be useful.

2.8 Proposition Let D, E be domains and let $f : D \to E$ be a function. Then f is Scott-continuous if and only if f is monotonic and for each $x \in D$ and $b \in \mathrm{approx}(f(x))$, there is $a \in \mathrm{approx}(x)$ such that $b \leq f(a)$.

Proof: Let f be Scott-continuous and let $b \in \mathrm{approx}(f(x))$. Then $b \leq f(x) = \sup f(\mathrm{approx}(x))$. By monotonicity, $f(\mathrm{approx}(x))$ is directed, and by compactness of b, there is some $f(a) \in f(\mathrm{approx}(x))$ with $b \leq f(a)$ as required.
Conversely, let $x \in D$. By monotonicity of f, $f(\mathrm{approx}(x))$ is directed, so $\sup f(\mathrm{approx}(x))$ exists. Now let $b \in \mathrm{approx}(f(x))$. Then there is $a \in \mathrm{approx}(x)$ with $b \leq f(a)$ by hypothesis, and therefore $f(x) = \sup \mathrm{approx}(f(x)) \leq \sup f(\mathrm{approx}(x))$ and by monotonicity $f(x) = \sup f(\mathrm{approx}(x))$. ■

2.9 Theorem (see [SLG94]) Let C, D be domains and let $f : D \to E$ be a function. Then f is Scott-continuous if and only if f is continuous with respect to the Scott topology $\mathcal{S}$.

Proof: Let $f : D \to E$ be Scott-continuous and choose a basic open set $\uparrow c$ for some $c \in E_c$. We have to show that $f^{-1}(\uparrow c)$ is open with respect to $\mathcal{S}$. Let $x \in f^{-1}(\uparrow c)$ and choose $y \geq x$. Then $c \leq f(x) \leq f(y)$ by monotonicity of f, hence $y \in f^{-1}(\uparrow c)$. By Proposition 2.8, there exists some $a \in \mathrm{approx}(x) \subseteq D_c$ such that $c \leq f(a)$ and therefore $a \in f^{-1}(\uparrow c)$ as required.
Conversely, let f be continuous with respect to $\mathcal{S}$, suppose $x \leq y$ and let $c \in \mathrm{approx}(f(x))$. Then $x \in f^{-1}(\uparrow c)$ and hence $y \in f^{-1}(\uparrow c)$ since $f^{-1}(\uparrow c)$ is open in $\mathcal{S}$. It follows that $c \in \mathrm{approx}(f(y))$, so we have $\mathrm{approx}(f(x)) \subseteq \mathrm{approx}(f(y))$ and therefore $f(x) \leq f(y)$, which shows monotonicity of f. Now let $c \in \mathrm{approx}(f(x))$. Then $x \in f^{-1}(\uparrow c)$ and by definition of $\mathcal{S}$, there is $a \in \mathrm{approx}(x)$ with $a \in f^{-1}(\uparrow c)$. Consequently $f(a) \in \uparrow c$ and $c \leq f(a)$ as required by Proposition 2.8. ■

2.2 Application to Definite Logic Programs

In order to order to apply the Knaster-Tarski Theorem to logic programming, we need to show that I_P is a domain. We will derive this using the following more general result.

2.10 Theorem (see [ACo80]) Let X be a set and let $\mathbf{2} = \{0, 1\}$.

(1) $\mathbf{2}^X$ is a domain with respect to set-inclusion. Its compact elements are exactly the finite subsets of X.

(2) Let $\mathbf{2}$ be endowed with the ordering $0 < 1$ and the Scott topology[2]. Then the product topology on $\mathbf{2}^X$ coincides with the Scott topology $\mathcal{S}$ on $\mathbf{2}^X$.

Proof: Statement (1) is immediate by definition. To see (2), we identify $\mathbf{2}^X$ with $\{\chi \mid \chi : X \to \mathbf{2}\}$ by identifying $A \subseteq X$ with the function χ, which takes the value 1 exactly on all elements in A. Let $\uparrow c$, c compact, be an arbitrary basic open set in $(\mathbf{2}^X, \mathcal{S})$. Now c can be identified with the function χ_c taking the value 1 exactly on the elements of c. The set $\uparrow c$ can then be identified with the set of functions taking the value 1 at least at the points where χ_c does. By definition of the product topology on $\mathbf{2}^X$, a basis is given by putting $\{1\}$ on finitely many coordinates, and $\{0, 1\} = \mathbf{2}$ on the remaining ones. As we have just seen, these basic open sets are exactly the sets of the form $\uparrow c$. So the two topologies have the same basis and must therefore coincide. ■

2.11 Corollary The set I_P, ordered by set-inclusion, is a domain, where the compact elements are exactly the finite subsets of B_P. Moreover, the Scott topology on I_P coincides with the product topology of $\mathbf{2}^{B_P}$, if $\mathbf{2}$ is endowed with the Scott topology.

Proof: As noted in the introduction, I_P can be identified with the product space $\prod_{A \in B_P} \mathbf{2} = \mathbf{2}^{B_P}$. The rest follows immediately from the previous theorem. ■

In order to apply the Knaster-Tarski Theorem it remains to show that the immediate consequence operator is Scott-continuous. It should be noted here that this is true only for definite logic programs. For normal logic programs, the immediate consequence operator is in general not monotonic, hence not Scott-continuous. For an example, check Program 3.

The following theorem is a main result for semantics of definite programs.

2.12 Theorem [Llo88] For any given definite logic program P, T_P is Scott-continuous.

Proof: Clearly, T_P is monotonic, so the image of every directed set in I_P is directed. Furthermore, we have for every directed subset X of I_P

$$\begin{aligned} A \in T_P(\sup X) &\iff \exists (A \leftarrow A_1, \ldots, A_k) \in \text{ground}(P) \text{ and } \{A_1, \ldots, A_k\} \subseteq \sup X \\ &\iff \exists (A \leftarrow A_1, \ldots, A_k) \in \text{ground}(P) \text{ and } \exists I \in X : \{A_1, \ldots, A_k\} \subseteq I \\ &\iff A \in T_P(I) \text{ for some } I \in X \\ &\iff A \in \sup T_P(X) \end{aligned}$$

[2] This space is known as the Sierpinski-space.

which proves the theorem. ∎

Thus, the Knaster-Tarski Theorem yields a procedure for finding the least fixed point M of T_P, which therefore is the least supported model for P. For an example, see Program 1.

It should be noted that in the case of a definite logic program P, M, as constructed above, coincides with the set of all logical consequences of P, which in turn coincides with the set of all ground atoms in $\mathcal{L}_P$ which are derivable from P via SLD-resolution (see [Llo88]). This result is well-known as the Kowalski-van Emden Theorem.

In the remaining section, we show that the Scott-topology on I_P can be recovered in a more natural way, using logical notions[3].

2.13 Definition (Positive atomic topology) (see [Sed95]) Let P be a logic program. The set $\{\mathcal{G}(A) \mid A \in B_P\}$ with $\mathcal{G}(A) = \{I \in I_P \mid A \in I\}$ is subbase of a topology, the *positive atomic topology* Q^+ on I_P.

Note that a basic open set in Q^+ is of the form $\mathcal{G}(A_1) \cap \cdots \cap \mathcal{G}(A_n)$, which we will write as $\mathcal{G}(A_1, \ldots, A_n)$. Since B_P is countable, it is immediate that Q^+ is second countable.

We have the following useful characterization of convergence in Q^+ which, as the remaining results of this section, can be found in [Sed95].

2.14 Proposition A sequence (I_n) converges in Q^+ to $I \in I_P$ if and only if every element of I is eventually an element of I_n[4].

Proof: Let $I_n \to I$ in Q^+ and $A \in I$. Since $\mathcal{G}(A)$ is a neighbourhood of I we have $I_n \in \mathcal{G}(A)$ eventually. So there exists $n_0 \in \mathbb{N}$ such that $I_n \in \mathcal{G}(A)$ for all $n \geq n_0$, which implies $A \in I_n$ for all $n \geq n_0$.
For the converse, suppose that the condition holds for (I_n) and I. Now choose a neighbourhood of I which, without loss of generality, is a basic neighbourhood, say $\mathcal{G}(A_1, \ldots, A_k)$. So we have $A_1, \ldots, A_k \in I$. By hypothesis, there is $n_0 \in \mathbb{N}$ such that $A_1, \ldots, A_k \in I_n$ for all $n \geq n_0$, thus $I_n \in \mathcal{G}(A_1, \ldots, A_k)$ for all $n \geq n_0$, which shows $I_n \to I$ in Q^+. ∎

2.15 Proposition The positive atomic topology Q^+ on I_P coincides with the Scott topology $\mathcal{S}$ on I_P.

Proof: Obviously, $\mathcal{G}(A_1, \ldots, A_k) = \uparrow\{A_1, \ldots, A_k\}$. So every basic open set in $\mathcal{S}$ is a basic open set in Q^+ and vice versa. ∎

[3] In fact, the results can be extended to a more general setting, using arbitrary preinterpretations. In the case of Herbrand preinterpretations, the positive atomic topology coincides with the positive query topology given in [BS89].

[4] For the more general setting, using arbitrary preinterpretations, the result still holds if "sequence I_n" is replaced by "net I_λ".

The following observation will be used in Section 5.

2.16 Proposition Let (I_n) be a sequence in I_P. Then the following hold.

(1) (I_n) has a greatest limit in Q^+, denoted by $\mathrm{gl}(I_n)$.

(2) $\mathrm{gl}(I_n) = \{A \in B_P \mid A \in I_n \text{ eventually}\}$.

(3) If (I_n) is eventually monotonic increasing, say $(I_k)_{k\geq k_0}$ is monotonic increasing, then $\mathrm{gl}(I_n) = \bigcup_{k\geq k_0} I_k$.

Proof: By Proposition 2.14, $\emptyset \in \{I \in I_P \mid I_n \to I\}$. Let $\mathrm{gl}(I_n) := \bigcup\{I \in I_P \mid I_n \to I\}$. By Proposition 2.14, $I_n \to \mathrm{gl}(I_n)$, and by construction of $\mathrm{gl}(I_n)$, it is the greatest limit of I_n. For the second statement, $\mathrm{gl}(I_n) \subseteq \{A \in B_P \mid A \in I_n \text{ eventually}\}$ is immediate, since $\mathrm{gl}(I_n)$ is a limit of I_n. For the converse, let $A \in I_n$ eventually. By Proposition 2.14, $I_n \to \{A\}$ in Q^+, and therefore $A \in \mathrm{gl}(I_n)$ by definition of $\mathrm{gl}(I_n)$. The third statement follows immediately from the second. ■

Summary

We have seen that every definite logic program has a least model, which is supported. This model can be interpreted as the intended meaning, or simply the semantics of P. The Kowalski-van Emden Theorem supports this view. So the issue of semantics of definite programs can be considered as resolved. The Knaster-Tarski Theorem yields a computational procedure for finding the semantics for P.

Although all computable functions can be implemented using definite programs, the syntax is very limited. Incorporating the use of negation in the syntax of logic programs, yielding normal logic programs, destroys monotonicity of the immediate consequence operator, thus making it impossible to use the Knaster-Tarski Theorem for finding supported models. So different approaches have to be found, and some of these will be studied in the following.

3 The Atomic Topology

In the previous section, we studied the Scott topology on I_P and applied it to logic programming semantics of definite programs. We have seen that the Scott topology is not appropriate for normal logic programs. In this section, we make use of the Cantor topology on I_P, introducing it via logical notions in Section 3.1 and applying it in Section 3.2.

3.1 The Atomic Topology Q

We define a natural topology on I_P and show that it conicides with the Cantor topology on the Cantor set in the unit-interval within the real line.

3.1 Definition (Atomic topology) (see [Sed95]) Let P be a logic program. The set $\{\mathcal{G}(A) \mid A \in B_P\} \cup \{\mathcal{G}(\neg A) \mid A \in B_P\}$, where $\mathcal{G}(A) = \{I \in I_P \mid A \in I\}$ and $\mathcal{G}(\neg A) = \{I \in I_P \mid A \notin I\}$, is subbase of a topology, the *atomic topology*[5] Q on I_P.

Note that the basic open sets of Q are of the form $\mathcal{G}(A_1) \cap \cdots \cap \mathcal{G}(A_k) \cap \mathcal{G}(\neg B_1) \cap \cdots \cap \mathcal{G}(\neg B_l)$, which we will write as $\mathcal{G}(A_1, \ldots, A_k, \neg B_1, \ldots, \neg B_l)$. Clearly, Q is second countable[6].

We give a characterization of convergence in Q.

3.2 Proposition (see [Sed95]) A sequence (I_n) converges in Q to $I \in I_P$ if and only if every element in I is eventually in I_n and every element not in I is eventually not in I_n[7].

Proof: Let $I_n \to I$ in Q and $A \in I$. Now $\mathcal{G}(A)$ is a neighbourhood of I and so $I_n \in \mathcal{G}(A)$ eventually. So there exists $n_0 \in \mathbb{N}$ such that $I_n \in \mathcal{G}(A)$ for all $n \geq n_0$, which implies $A \in I_n$ for all $n \geq n_0$. For $A \notin I$ choose $\mathcal{G}(\neg A)$ as a neighbourhood of I. Since $I_n \to I$ in Q, we have $n_0 \in \mathbb{N}$ with $I_n \in \mathcal{G}(\neg A)$ for all $n \geq n_0$. Hence $A \notin I_n$ for all $n \geq n_0$ as required.

For the converse, suppose that the condition holds for (I_n) and I. Now choose a neighbourhood of I which, without loss of generality, is a basic neighbourhood, say $\mathcal{G}(A_1, \ldots, A_k, \neg B_1, \ldots, \neg B_l)$. So we have $A_1, \ldots, A_k \in I$ and $B_1, \ldots, B_l \notin I$. By hypothesis, there is $n_0 \in \mathbb{N}$ such that $A_1, \ldots, A_k \in I_n$ and $B_1, \ldots, B_l \notin I_n$ for all $n \geq n_0$, thus $I_n \in \mathcal{G}(A_1, \ldots, A_k, \neg B_1, \ldots, \neg B_l)$ for all $n \geq n_0$, which shows $I_n \to I$ in Q. ■

We proceed with studying properties of the atomic topology. The following theorem will be useful.

[5] The topology can be extended to a more general setting, using arbitrary preinterpretations. In the case of Herbrand preinterpretations, the atomic topology coincides with the query topology given in [BS89].

[6] which is not true for arbitrary preinterpretations whose domain is not countable.

[7] For the more general setting, using arbitrary preinterpretations, the result still holds if "sequence I_n" is replaced by "net I_λ".

3.3 Theorem (see [Wil70]) Let X be a finite or countable set, let $\mathbf{2} = \{0, 1\}$ be endowed with the discrete topology and let $\mathcal{T}$ denote the product topology on $\mathbf{2}^X$.

(1) $(\mathbf{2}^X, \mathcal{T})$ is a totally disconnected compact Hausdorff space.

(2) $\mathcal{T}$ is second countable and metrizable.

(3) $(\mathbf{2}^X, \mathcal{T})$ is homeomorphic to the Cantor set in the closed unit interval within the real line if X is infinite, i.e. if the first order language underlying P contains at least one function symbol.

Proof: Note that the discrete topology on $\mathbf{2}$ is totally disconnected, compact, Hausdorff, second countable and metrizable. Since X is countable, all these properties carry over to the product topology by well-known topological results. It is equally well-known that $(\mathbf{2}^X, \mathcal{T})$ is homeomorphic to the Cantor set if X is infinite. For details, check [Wil70], Theorems 29.3, 17.8, 13.8, 16.2, 22.3, and Corollary 30.5, respectively. ■

3.4 Proposition (see [Sed95]) The atomic topology on I_P coincides with the product topology on $\mathbf{2}^{B_P}$, where $\mathbf{2} = \{0, 1\}$ is endowed with the discrete topology.

Proof: As noted in the introduction, $\mathbf{2}^{B_P}$ can be identified with $\{\chi \mid \chi : B_P \to \mathbf{2}\}$ by identifying each $I \in I_P$ with the function χ_I, mapping all $A \in I$ to 1 and all $A \notin I$ to 0 and vice versa.
Now let I_n be a sequence in I_P which converges in Q to some $I \in I_P$. If $A \in I$, then $\chi_I(A) = 1$ and by Proposition 3.2, A is eventually in I_n, i.e. $\chi_{I_n}(A) = 1$ eventually. If $A \notin I$, then $\chi_I(A) = 0$, and by Proposition 3.2, $A \notin I_n$ eventually, hence $\chi_{I_n}(A) = 0$ eventually. So χ_{I_n} converges pointwise to χ_I. The above argument reverses, and we see, that Q is in fact the topology of pointwise convergence in $\mathbf{2}^{B_P}$. By well-known topological results, this topology coincides with the above product topology (see [Wil70, Definition 42.1 and Theorem 42.2]). ■

Together, we get the following:

3.5 Theorem (see [Sed95]) (I_P, Q) is a totally disconnected compact Hausdorff space which is second countable[8] and metrizable[8]. It is homeomorphic to the Cantor set on the real line, if B_P is infinite.

3.2 Models and the Atomic Topology

In this section, we study the connections between (supported) models and convergence and continuity in Q, which will support the view that the atomic topology is a very suitable topology for the purposes of the semantics of normal logic programs.

[8] In the more general setting, these are true if and only if the domain of the preinterpretation is countable.

The following theorem states that if an orbit of the immediate consequence operator of some normal logic program P converges in Q, its limit is already a model for P. The result is new and will successfully be applied in Section 7.

3.6 Theorem Let P be a normal logic program.

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

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

Proof: Suppose $T_P^n(I) \to M$ in Q for some $I \in I_P$. We have to show that $T_P(M) \subseteq M$. Let $A \in T_P(M)$. By definition of T_P, there exists a ground instance $A \leftarrow A_1, \dots, A_{k_1}, \neg B_1, \dots, \neg B_{l_1}$ of a clause in P with $A_k \in M$ and $B_l \notin M$ for $k = 1, \dots, k_1$, $l = 1, \dots, l_1$. By Proposition 3.2, there is an $n_0 \in \mathbb{N}$, such that for all $n \geq n_0$, $A_k \in T_P^n(I)$ and $B_l \notin T_P^n(I)$ for all k, l. By definition of T_P and the above clause we have that $A \in T_P^m(I)$ for all $m \geq n_0 + 1$. Hence, $A \in T_P^n(I)$ eventually and therefore, by Proposition 3.2 again, $A \in M$, which proves the first statement.

Now, if M is a supported model for P, then $(T_P^n(M))$ is constant with value M, so the second statement is trivially true. ■

3.7 Remark In the situation of Theorem 3.6, M is in general not supported. See Program 8 as an example.

Let P be a normal logic program and let $I \in I_P$ be such that the sequence $(T_P^n(I))$ converges in Q to some $M \in I_P$. Then by Theorem 3.6, M is a model for P. If, furthermore, T_P is continuous in Q, or at least continuous at M, then $M = \lim T_P^{n+1}(I) = \lim T_P(T_P^n(I)) = T_P(\lim T_P^n(I)) = T_P(M)$. So M is a supported model in this case.

Continuity of the immediate consequence operator is studied in detail in [Sed95]. We will not discuss this here, since supported models can be obtained by much weaker assumptions. The following results give some semi-syntactic conditions on logic programs to ensure that the limit of a converging orbit under T_P is a supported model.

The following theorem is new. Recall that a variable symbol in a program clause is called *local* if it only appears in the body of that clause.

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

Proof: We have to show that $M \subseteq T_P(M)$. So let $A \in M$. By convergence in Q and Proposition 3.2, there exists $n_0 \in \mathbb{N}$ such that $A \in T_P^n(I_0)$ for all $n \geq n_0$. By hypothesis, there are only finitely many clauses in ground(P) with head A. Let C_0 be the (finite) set of all atoms occuring in positive body literals and D_0 the (finite) set of all atoms occuring in negative body literals of those clauses. Let $C_1 = C_0 \cap M$ and $D_1 = D_0 \setminus M$. Since $I_n \to M$ in Q, there is an $n_1 \in \mathbb{N}$ such that $C_1 \subseteq I_n$ and $D_1 \subseteq B_P \setminus I_n$ for all $n \geq n_1$. Since $A \in T_P(I_{\max\{n_0,n_1\}})$, there is a clause $A \leftarrow A_1, \dots, A_{k_1}, \neg B_1, \dots, \neg B_{l_1}$ in ground(P) with $A_k \in C_1 \subseteq M$ and $B_l \notin D_1 \subseteq B_P \setminus M$ for $k = 1, \dots, k_1$, $l = 1, \dots, l_1$. Hence $A \in T_P(M)$ as required. ■

3.9 Remark In the situation of the previous theorem, M need not be minimal. For an example, see Program 6.

The remaining results of this section can be found in [Sed95], although we use the new result Theorem 3.8 to prove them.

3.10 Corollary Let P be a normal logic program and let $I_0 \in I_P$ be such that the sequence $T_P^n(I_0)$ converges in Q to some $M \in I_P$. If P contains no local variables, then M is a supported model for P[9].

Proof: Immediately by the previous theorem. ■

3.11 Theorem Let P be a normal logic program and let $I_0 \in I_P$ be such that the sequence $T_P^n(I_0)$ converges in Q to some $M \in I_P$. If for every $A \notin T_P(M)$ no clause whose head matches A contains a local variable, then M is a supported model[10].

Proof: We have to show that $M \subseteq T_P(M)$. Suppose there is some $A \in M \setminus T_P(M)$. Then by the same argument as in the proof of Theorem 3.8 we get $A \in T_P(M)$, a contradiction. ■

Summary

We have seen that the Cantor topology Q on I_P can be defined by using logical notions. Converging orbits in Q always yield models as their limits, and some semi-syntactical conditions for ensuring that the model in question is supported were given.

Problems

Problem 1 Find neccessary and sufficent syntactic conditions for convergence in Q of orbits of the immediate consequence operator.

Problem 2 In the situation of Theorem 3.8, find neccessary and sufficient conditions to ensure that M is a minimal model for P.

[9]In fact, T_P is continuous in this case, see [Sed95, Corollary 6]

[10]In fact, T_P is continuous at M in this case, see [Sed95, Theorem 7].

Problem 3 Let $I \in I_P$. Consider the sequence (I_k) defined by $I_0 := I$ and $I_{k+1} := \lim T_P^n(I_k)$. When is this construction possible? Does (I_k) converge in Q? Does it become stable after finitely many steps? Is the limit a (supported) model?

4 Metric and Extended Metric Spaces

In this section, we will apply the Banach Contraction Mapping Theorem and the closely related theorem of Priess-Crampe and Ribenboim to logic programming semantics. We will show that every domain can be viewed as an extended ultrametric space, and that every strictly level-decreasing program has a unique supported model.

4.1 Domains as Extended Ultrametric Spaces

Recall the Banach Contraction Mapping Theorem.

4.1 Theorem (Banach) Let (X, d) be a complete metric space and let $f : X \to X$ be a contraction. Then f has a unique fixed point x. Furthermore, $x = \lim f^n(y)$ for every $y \in X$.

The main idea of the proof is that the orbit of an arbitrary point under f is a Cauchy sequence and therefore converges to some point, which turns out to be the only fixed point of f. A generalization, Theorem 5.5, will be proven later in detail.

Trying to apply the Banach Contraction Mapping Theorem in the area of programming semantics, the general problem arises that the spaces considered there are in general not Hausdorff. Furthermore, the T_P-operator considered in logic programming usually does not have a unique fixed point. So the theorem is only of limited use for our setting. We will see later that the theorem can be generalized to quasi-metrics, which are in general not Hausdorff and are much better suited for our purpose.

Nevertheless, as we show next, some results can be gained by applying a variant of the Banach Contraction Mapping Theorem, due to Priess-Crampe and Ribenboim, where we allow the image set of the distance function to be different from the real numbers, but require other stronger conditions on the space in question. We introduce these spaces next.

4.2 Definition (extended ultrametric space) Let X be a set and let Γ be a partial order with least element 0. We call (X, d) an *extended ultrametric space*[11] if $d : X \times X \to \Gamma$ is a function such that for all $x, y, z \in X$

(1) $d(x, y) = 0$ if and only if $x = y$,

(2) $d(x, y) = d(y, x)$, and

(3) if $d(x, y), d(y, z) \leq \gamma$ 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 in X. An extended ultrametric space is called *spherically complete* if for any chain $(\mathcal{C}, \subseteq)$ of balls in X, $\bigcap \mathcal{C} \neq \emptyset$.

A function $f : X \to X$ is called

[11] In [PR97], these spaces are called *generalized ultrametric spaces*, but we need this notion in Sections 5 and 6 for a different class of spaces.

(1) *non-expanding*[12] if $d(f(x), f(y)) \leq d(x, y)$ for all $x, y \in X$,

(2) *strictly contracting on orbits* if $d(f^2(x), f(x)) < d(f(x), x)$ for every $x \in X$ with $x \neq f(x)$, and

(3) *strictly contracting* if $d(f(x), f(y)) < d(x, y)$ for all $x, y \in X$ with $x \neq y$.

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

4.3 Lemma Let (X, d, Γ) be an extended ultrametric space. For $\alpha, \beta \in \Gamma$ and $x, y \in X$ the following statements hold.

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

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

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

Proof: Let $a \in B_\alpha(x)$ and $b \in B_\alpha(x) \cap B_\beta(y)$. 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 $d(a, y) \leq \beta$, hence $a \in B_\beta(y)$, which proves the first statement. The second follows by symmetry and the third by replacing α by $d(x, y)$. ■

The following theorem was given in [PR97] in a more general form.

4.4 Theorem (Priess-Crampe and Ribenboim) Let (X, d) be a spherically complete extended ultrametric space and let $f : X \to X$ be non-expanding and strictly contracting on orbits. Then f has a fixed point. Moreover, if f is strictly contracting on X, then f has a unique fixed point.

Proof: Assume that f has no fixed point. Then $d(x, f(x)) \neq 0$ for all $x \in X$. We define the set $\mathcal{B} := \{B_{d(x,f(x))}(x) \mid x \in X\}$. Now let $\mathcal{C}$ be a maximal chain in $\mathcal{B}$. Since X is spherically complete, there exists $z \in \bigcap \mathcal{C}$. We show, that $B_{d(z,f(z))} \subseteq \bigcap \mathcal{C}$. Let $B_{d(x,f(x))}(x) \in \mathcal{C}$. Since $z \in B_{d(x,f(x))}(x)$, we get $d(z, x) \leq d(x, f(x))$ and $d(z, f(x)) \leq d(x, f(x))$. By non-expansiveness of f, we get $d(f(z), f(x)) \leq d(z, x) \leq d(x, f(x))$. It follows, that $d(z, f(z)) \leq d(x, f(x))$ and therefore $B_{d(z,f(z))}(z) \subseteq B_{d(x,f(x))}(x)$ by Lemma 4.3. Since x was chosen arbitrarily, $B_{d(z,f(z))}(z) \subseteq \bigcap \mathcal{C}$.

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 4.3, 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 $\mathcal{C}$. So 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. ■

[12] In [PR97], these functions were called *contractive*.

Note that the above given proof is not constructive, so it does not indicate a means by which one can actually find a fixed point.

In order to apply this result, we show first how every domain can be viewed as a spherically complete extended ultrametric space. For some $\gamma \in \Omega$ (as defined in Section 1), let Γ_γ be the set $\{2^{-\alpha} \mid \alpha < \gamma\}$ of symbols $2^{-\alpha}$ with ordering $2^{-\alpha} < 2^{-\beta}$ if and only if $\beta < \alpha$.

4.5 Definition (see [SH97c]) Let D be a domain and $r : D_c \to \gamma$ a function, called a *rank function*, and denote $2^{-\gamma}$ by 0. Define $d_r : D \times D \to \Gamma_{\gamma+1}$ by

$$d_r(x, y) := \inf\{2^{-\alpha} \mid c \leq x \text{ if and only if } c \leq y \text{ for every } c \in D_c \text{ with } r(c) < \alpha\}.$$

Then (D, d_r) is called the extended ultrametric space *induced by* r.

It is straightforward to see, that (D, d_r) is indeed an extended ultrametric space.

We proceed to show, that (D, d_r) is spherically complete. For every extended ultrametric with image in Γ_α, we will denote the ball $B_{2^{-\alpha}}(x)$ in the following by $B_\alpha(x)$.

4.6 Lemma (see [SH97c]) Let $B_\alpha(x) \subseteq B_\beta(y)$ (so $\beta \leq \alpha$). Then the following statements hold.

(1) $\{c \in \text{approx}(x) \mid r(c) \leq \beta\} = \{c \in \text{approx}(y) \mid r(c) \leq \beta\}$.

(2) $B_\alpha := \sup\{c \in \text{approx}(x) \mid r(c) \leq \alpha\}$ and $B_\beta := \sup\{c \in \text{approx}(y) \mid r(c) \leq \beta\}$ both exist.

(3) $B_\beta \leq B_\alpha$.

Proof: Since $d_r(x, y) \leq 2^{-\beta}$, the first statement follows immediately from the definition of d_r. The second statement follows from the fact that every domain is consistently complete by Proposition 2.5. The third statement follows from the observation that $B_\beta = \sup\{c \in \text{approx}(y) \mid r(c) \leq \beta\} = \sup\{c \in \text{approx}(x) \mid r(c) \leq \beta\} \leq \sup\{c \in \text{approx}(x) \mid r(c) \leq \alpha\} = B_\alpha$. ∎

4.7 Theorem (see [SH97c]) (D, d_r) is spherically complete.

Proof: By the previous lemma, every chain $(B_\alpha(x_\alpha))$ of balls in D gives rise to a chain (B_α) in D in reverse order. Let $B := \sup B_\alpha$. Now let $B_\alpha(x)$ be an arbitrary ball in the chain. It suffices to show that $B \in B_\alpha(x)$. Since $B_\alpha \in B_\alpha(x)$, we have $d_r(B_\alpha, x) \leq 2^{-\alpha}$, and since d_r is an ultrametric, it remains to show that $d_r(B, B_\alpha) \leq 2^{-\alpha}$. For every $c \leq B_\alpha$, we have $c \leq B$ by construction of B. Now let $c \leq B$ with $c \in D_c$ and $r(c) < \alpha$. We have to show that $c \leq B_\alpha$. Since D is a domain, hence an algebraic cpo, there exists B_β in the chain with $c \leq B_\beta$. Now suppose $B_\beta \geq B_\alpha$ (otherwise $c \leq B_\alpha$ immediately). Then by the above lemma and the fact that the collection $(B_\alpha(x_\alpha))$ is a chain, we have $B_\beta(x_\beta) \subseteq B_\alpha(x_\alpha)$ and therefore $c \in \{c \in \text{approx}(x_\beta) \mid r(c) \leq \alpha\} = \{c \in \text{approx}(x_\alpha) \mid r(c) \leq \alpha\}$. Since B_α is the supremum of the right-hand set, $c \leq B_\alpha$. ∎

It should be noted that we needed both algebraicity and consistent completeness of domains to prove the previous theorem.

We apply this result to logic programming. We next introduce level mappings on I_P, which will be used for defining rank functions. Level mappings are one of the central notions in the sequel and will often be used later on. For the following, we denote the set of all finite subsets of I_P, which is the set of all compact elements in I_P, by I_c.

4.8 Definition (level mapping) Let P be a normal logic program and let $\gamma \in \Omega$. A mapping $l : B_P \to \gamma$ is called a *level mapping*. We call l an ω-level mapping if $\gamma = \omega$. We set $\mathcal{L}_\alpha := \{A \in B_P \mid l(A) < \alpha\}$ for $\alpha \leq \gamma$ and $\mathcal{L}_0 = \emptyset$. An ω-level mapping is called a *finite level mapping* if $\mathcal{L}_n$ is finite for every $n \in \mathbb{N}$.

We define the rank function *induced by* the level mapping l by $r(I) := \max\{l(A) \mid A \in I\}$ for every $I \in I_c$. An extended ultrametric obtained by such a rank function will further be denoted by d_l.

The following proposition makes calculation of distances easier.

4.9 Proposition Let P be a normal logic program, let l be a level mapping for P and let $I, J \in I_P$. Then $d_l(I, J) = \inf\{2^{-\alpha} \mid I \cap \mathcal{L}_\alpha = J \cap \mathcal{L}_\alpha\}$.

Proof: Immediate by the observation that for every $I \in I_P$, $I = \sup\{\{A\} \mid A \in I\}$. ■

4.2 Application to Strictly Level-decreasing Programs

In order to apply the above observations, we introduce a class of programs, called strictly level-decreasing programs, and the closely related class of semi-strictly level-decreasing programs. A subclass of the second will further be studied in Section 7.2.

4.10 Definition (see [SH97c]) Let P be a normal logic program. We call P

(1) *level-decreasing (with respect to l)* if there exists a level mapping l such that for every clause $H \leftarrow B_1, \ldots, B_{n_1}, \neg C_1, \ldots, \neg C_{n_2}$ in ground(P) $l(B_i) \leq l(H)$ and $l(C_j) \leq l(H)$ hold for every $i = 1, \ldots, n_1$ and $j = 1, \ldots, n_2$.

(2) *strictly level-decreasing (with respect to l)* if there exists a level mapping l such that for every clause $H \leftarrow B_1, \ldots, B_{n_1}, \neg C_1, \ldots, \neg C_{n_2}$ in ground(P) $l(B_i) < l(H)$ and $l(C_j) < l(H)$ hold for every $i = 1, \ldots, n_1$ and $j = 1, \ldots, n_2$.

(3) *semi-strictly level-decreasing (with respect to l)* if there exists a level mapping l such that for every clause $H \leftarrow B_1, \ldots, B_{n_1}, \neg C_1, \ldots, \neg C_{n_2}$ in ground(P) $l(B_i) \leq l(H)$ and $l(C_j) < l(H)$ hold for every $i = 1, \ldots, n_1$ and $j = 1, \ldots, n_2$.

The definition of level decreasing programs is too general to be of any use, even if l is an ω-level mapping. See Program 2 as an example. In order to apply Theorem 4.4, T_P must be strictly contracting.

4.11 Theorem (see [SH97c]) Let P be a strictly level-decreasing program with respect to a level mapping l. Then T_P is strictly contracting on (I_P, d_l).

Proof: Let $I_1, I_2 \in I_P$ with $d(I_1, I_2) = 2^{-\alpha}$.

(1) Let $\alpha = 0$, so I_1 and I_2 differ on some element of B_P with level 0. Let $A \in T_P(I_1)$ with $l(A) = 0$. Since P is strictly level-decreasing, A must be the head of a clause in ground(P) and so $A \in T_P(I_2)$. By the same argument, if $A \in T_P(I_2)$ with $l(A) = 0$, then $A \in T_P(I_1)$. So $T_P(I_1) \cap \mathcal{L}_1 = T_P(I_2) \cap \mathcal{L}_1$, and it follows that

$$d(T_P(I_1), T_P(I_2)) \leq 2^{-1} < 2^{-0} = d(I_1, I_2)$$

as required.

(2) Let $\alpha > 0$, so I_1 and I_2 differ on some element of B_P with level α but agree on all ground atoms of lower level. Let $A \in T_P(I_1)$ with $l(A) \leq \alpha$. Then there is a clause $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1} \in$ ground(P), where $k_1, l_1 \geq 0$, with $A_k \in I_1$ and $B_l \notin I_1$ for all $k = 1, \ldots, k_1$, $l = 1, \ldots, l_1$. Since P is strictly level-decreasing and $I_1 \cap \mathcal{L}_\alpha = I_2 \cap \mathcal{L}_\alpha$, it follows that $A_k \in I_2$ and $B_l \notin I_2$ for $k = 1, \ldots, k_1$, $l = 1, \ldots, l_1$. Therefore, $A \in T_P(I_2)$. By the same argument, if $A \in T_P(I_2)$ with $l(A) = 0$, then $A \in T_P(I_1)$. So $T_P(I_1) \cap \mathcal{L}_{\alpha+1} = T_P(I_2) \cap \mathcal{L}_{\alpha+1}$, and it follows that

$$d(T_P(I_1), T_P(I_2)) \leq 2^{-(\alpha+1)} < 2^{-\alpha} = d(I_1, I_2)$$

as required. ■

4.12 Theorem (see [SH97c]) Let P be a strictly level-decreasing logic program. Then T_P has a unique fixed point and hence P has a unique supported model.

Proof: Immediate by Theorem 4.4 and the previous theorems. ■

For an application of the above result, see Program 4. Note that Theorem 4.4 only yields the existence of a unique model for strictly level-decreasing programs. Its proof does not provide a method for actually finding it.

4.13 Remark The above result does not hold for semi-strictly level-decreasing progams. See Program 10 as an example. But this class will be further studied in Section 7.2, as mentioned already.

It is interesting to further study the above observations in the special case when l is an ω-level mapping. This ultrametric was studied in [Fit94], and we repeat the definition for this case.

4.14 Definition Let P be a normal logic program and $l : B_P \to \omega$ an ω-level mapping. Then l induces an ultrametric ϱ on I_P by

$$\varrho(I_1, I_2) := \inf\{2^{-n} \mid \mathcal{L}_n \cap I_1 = \mathcal{L}_n \cap I_2\}.$$

The following proposition exhibits the strong connection between ϱ and Q.

4.15 Proposition (see [Sed97]) If a sequence (I_n) in (I_P, ϱ) is a Cauchy sequence, then every $A \in B_P$ is either eventually in I_n or eventually not in I_n. Furthermore, (I_P, ϱ) is complete. If the level mapping underlying ϱ is finite, then ϱ induces the atomic topology on I_P.

Proof: Let (I_n) be a Cauchy sequence with respect to ϱ and let $A \in B_P$ with level $l(A) = k$. By definition of Cauchy sequence, there is $k_0 \in \mathbb{N}$ such that $d(I_l, I_m) < 2^{-k}$ for all $l, m \geq k_0$. So all members of the sequence $(I_n)_{n \geq k_0}$ agree on all atoms of level k: Hence $A \in I_n$ for all $n \geq k_0$ or $A \notin I_n$ for all $n \geq k_0$. It follows that (I_n) converges to $\{A \in B_P \mid A \in I_n \text{ eventually}\}$ and hence the metric is complete. Now suppose the level mapping l is finite. For the third statement, it remains to show that if (I_n) converges in Q, then it is a Cauchy sequence. Choose $k \in \mathbb{N}$. By Proposition 3.2, and since the level mapping is finite, there is a $k_0 \in \mathbb{N}$ such that all members of the sequence $(I_n)_{n \geq k_0}$ agree on $\mathcal{L}_k$. So for all $l, m \geq k_0$, $d(I_l, I_m) < 2^{-k}$ as required. ■

Only one more step is missing before we can apply the Banach Contraction Mapping Theorem:

4.16 Theorem (see [SH97c]) For every program P which is strictly level-decreasing with respect to an ω-level mapping, T_P is a contraction with contractivity factor $\frac{1}{2}$. Hence, P has a unique supported model, obtained as the limit in Q of the orbit of $\emptyset$ under T_P.

Proof: The proof that T_P is a contraction with contractivity factor $\frac{1}{2}$ is almost exactly the proof of Proposition 4.11, and we therefore omit the details. By the previous proposition, Q is finer than the topology underlying ϱ, so by Theorem 4.1 the unique fixed point of T_P is obtained as stated. ■

For an application of this result see Program 3. Note that the proof of the Banach Contraction Mapping Theorem (even the proof of the Rutten-Smyth Theorem) provides a construction of the above fixed point.

Programs which are strictly level decreasing logic programs with respect to an ω-level mapping are *acceptable*, as defined in [Fit94, Definition 7.1]. In the cited paper, a unique fixed point for acceptable programs is derived by iterating the single-step operator, using a different metric from ours.

Summary

We have seen that both the Banach Contraction Mapping Theorem and the theorem of Priess-Crampe and Ribenboim can be applied to logic programming semantics. In fact, it was possible to show that strictly level-decreasing programs have a unique supported model. Therefore, all the standard approaches to logic programming semantics coincide for these programs. Furthermore, we have seen that there is a relationship between domains and extended ultrametric spaces comparable with the relationship between domains and quasi-metrics as studied in [Smy91].

Problems

Problem 4 To what extent can the construction of extended ultrametric spaces out of domains, as done in Section 4.1, be reversed?

Problem 5 Examine the relationships between domains and extended ultrametric spaces.

Problem 6 To what extent can Theorem 4.12 be reversed?

Problem 7 Try to find a contructive proof of Theorem 4.4 in order to find a fixed point of the function given there in the hypothesis.

5 Quasi-metric Spaces

We have seen in the previous sections, that both the Knaster-Tarski Theorem 2.3 and the Banach Contraction Mapping Theorem 4.1 can be applied in the area of logic programming semantics. We give a generalization of both of these results, the Rutten-Smyth Theorem 5.5, as stated in [Rut96] and show how this theorem can be applied to our setting. A slightly more general version of Theorem 5.5 is given in [Smy87].

5.1 Domains as Quasi-ultrametric Spaces

We first introduce generalized metric spaces. A discussion of the following notions can be found in [Rut96] and [Smy91].

5.1 Definition (gms) A set X together with a function $d : X \times X \to \mathbb{R} \cup \{\infty\}$ is called a *generalized metric space* (*gms*) if for all $x, y, z \in X$

(1) $d(x, x) = 0$ and

(2) $d(x, z) \leq d(x, y) + d(y, z)$.

If, furthermore, $d(x, y) = d(y, x) = 0$ implies $x = y$, then (X, d) is called a *quasi-metric space* (*qms*). A gms in which the strong triangle inequality $d(x, y) \leq \max\{d(x, z), d(z, y)\}$ holds for all $x, y, z \in X$, is called a generalized *ultra*metric space. Consequently, a generalized ultrametric space which is a quasi-metric space is called a *quasi-ultrametric space*. A sequence (x_n) in X is a (*forward-*) *Cauchy-sequence* (*CS*) if, for all $\varepsilon > 0$, there exists $n_0 \in \mathbb{N}$ such that for all $n \geq m \geq n_0$, $d(x_m, x_n) < \varepsilon$. A CS (x_n) *converges to* $x \in X$ ($x_n \to x$, $\lim x_n = x$) if, for all $y \in X$, $d(x, y) = \lim d(x_n, y)$. Finally, X is called *CS-complete* if every CS in X converges.

Note that limits of CSs need not be unique. If X is a qms, however, uniqueness of limits does hold. Indeed, for $x_n \to x$ and $x_n \to y$, we get $d(x, y) = \lim d(x_n, y) = d(y, y) = 0$ and $d(y, x) = \lim d(x_n, x) = d(x, x) = 0$, which shows $x = y$.

5.2 Example Let $(X, \leq)$ be a partial order. Define a function $d_\leq : X \times X \to \mathbb{R}$ by

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

Then it is easily checked that $(X, d_\leq)$ is a quasi-ultrametric space, and $d_\leq$ is called the *discrete quasimetric* on X.

5.3 Remark If (X, d) is a gms, then (X, d^*) is a metric space, where $d^*(x, y) := \max\{d(x, y), d(y, x)\}$.

5.4 Definition (CS-continuous, contractive, non-expanding) Let X be a gms. A function $f : X \to X$ is called

(1) *CS-continuous* if, for all CSs (x_n) in X with $\lim x_n = x$, $(f(x_n))$ is a CS and $\lim f(x_n) = f(x)$,

(2) *non-expanding* if $d(f(x), f(y)) \leq d(x, y)$ for all $x, y \in X$, and

(3) *contractive* if there exists some $0 \leq c < 1$ such that $d(f(x), f(y)) \leq c \cdot d(x, y)$ for all $x, y \in X$.

Given a qms (X, d), d induces a partial order $\leq_d$ on X by setting $x \leq_d y$ if and only if $d(x, y) = 0$. Note that $\leq_{d_\leq}$ and $\leq$ coincide, so the above construction reverses Example 5.2.

5.5 Theorem (Rutten-Smyth) (see [Rut96]) Let $X \neq \emptyset$ be a complete qms and let $f : X \to X$ be non-expanding.

(1) If f is CS-continuous and there exists $x \in X$ with $x \leq_d f(x)$, then f has a fixed point, and this fixed point is least above x with respect to $\leq_d$.

(2) If f is CS-continuous and contractive, then f has a unique fixed point.

Moreover, in both cases the fixed point can be obtained as the limit of the CS $(f^n(x))$, where in (1) x is the given point, and in (2) x can be chosen arbitrarily.

Proof: (1): We have $d(f^n(x), f^{n+1}(x)) \leq d(x, f(x)) = 0$ and $d(f^n(x), f^{n+k}(x)) \leq \sum_{i=0}^{k-1} d(f^{n+i}(x), f^{n+i+1}(x)) = 0$. Hence $(f^n(x))$ is a CS and has a unique limit y. Since $f(y) = f(\lim f^n(x)) = \lim f(f^n(x)) = \lim f^n(x) = y$, y is a fixed point of f. Now let $z = f(z)$ be a fixed point of f with $x \leq_d z$. Then $d(y, z) = \lim d(f^n(x), z) = 0$, since $d(f^n(x), f^n(z)) \leq d(x, z) = 0$. Hence $y \leq_d z$.

(2): Let $x \in X$ be chosen arbitrarily. We have $d(f^n(x), f^{n+k}(x)) = d(f^n(x), f^n(f^k(x))) \leq c^n d(x, f^k(x)) \leq c^n \sum_{i=0}^{k-1} d(f^i(x), f^{i+1}(x)) \leq c^n \sum_{i=0}^{k-1} c^i d(x, f(x)) \leq \frac{c^n}{1-c} d(x, f(x))$, which converges to 0 as $n \to \infty$. So $(f^n(x))$ is a CS and has a unique limit $y = \lim f^{n+1}(x) = \lim f(f^n(x)) = f(\lim f^n(x)) = f(y)$. Now if $z = f(z)$, we have $d(y, z) = d(f(y), f(z)) \leq c \cdot d(y, z)$, hence $d(y, z) = 0$. Similarly, $d(z, y) = 0$, and hence $y = z$. ■

Part (1) of Theorem 5.5 generalizes the Knaster-Tarski Theorem 2.3 by virtue of Example 5.2. Part (2) generalizes the Banach Contraction Mapping Theorem 4.1.

In order to apply this theorem, we have to cast the space I_P into a quasi-metric space. One possible way was already shown in Example 5.2, and this approach will enable us to recover the standard semantics of definite logic programs via the Rutten-Smyth Theorem, as was done in [Sed97].

5.6 Proposition (see [Sed97]) Let $d = d_\subseteq$ be the discrete quasimetric on I_P. Then the following statements hold.

(1) (I_n) is a CS if and only if I_n is eventually monotonic increasing.

(2) (I_P, d) is complete.

(3) For every CS I_n, $\lim I_n = \mathrm{gl}(I_n)$.

Proof: The first statement follows immediately from the definition of d. Now let (I_n) be a CS in I_P. Then (I_n) is eventually monotonic increasing with greatest limit $\mathrm{gl}(I_n)$ in Q^+. It remains to show that $\mathrm{gl}(I_n)$ is the limit of the CS (I_n) with respect to d. Now let $k_0 \in \mathbb{N}$ such that $(I_k)_{k \geq k_0}$ is monotonic increasing. Then we have to show that for all $J \in I_P$, $d(\mathrm{gl}(I_n), J) = \lim d(I_n, J)$. Suppose $\mathrm{gl}(I_n) \subseteq J$. Then by Proposition 2.16, $I_k \subseteq \mathrm{gl}(I_n)$ for all $k \geq k_0$ and hence $I_n \subseteq J$ eventually. Therefore, $d(\mathrm{gl}(I_n), J) = 0 = \lim d(I_n, J)$ in this case. Suppose $\mathrm{gl}(I_n) \not\subseteq J$. Then there is $A \in J$ with $A \notin I_n$ eventually by Proposition 2.16 and the fact that I_n is eventually monotonic increasing. It follows that $d(I_n, J) = 1$ eventually as required. ■

We need one more step to apply the Rutten-Smyth Theorem.

5.7 Theorem (see [Sed97]) Let P be a definite logic program and $d = d_{\leq}$ be the discrete quasimetric on I_P. Then the following statements hold.

(1) (I_P, d) is a complete quasi-ultrametric space.

(2) $T_P : I_P \longrightarrow I_P$ is CS-continuous and non-expanding.

(3) $\emptyset \in I_P$ and $d(\emptyset, T_P(\emptyset)) = 0$.

Proof: Point (3) is trivially true, and so by Proposition 5.6 it only remains to prove point (2). Non-expansiveness of T_P follows immediately from monotonicity of T_P. We now show CS-continuity: Let (I_n) be a CS in I_P with $\lim I_n = \mathrm{gl}(I_n)$. Since (I_n) is eventually increasing, there is $k_0 \in \mathbb{N}$ such that $(I_k)_{k \geq k_0}$ is monotonic increasing. Since T_P is monotonic, the sequence $(T_P(I_k))_{k \geq k_0}$ is monotonic increasing, so $(T_P(I_n))$ is a CS and has a limit $\mathrm{gl}(T_P(I_n))$ by Proposition 5.6. We have to show that $\mathrm{gl}(T_P(I_n)) = T_P(\mathrm{gl}(I_n))$. By Propositon 2.16, $\mathrm{gl}(I_n) = \bigcup_{k \geq k_0} I_k$ and $\mathrm{gl}(T_P(I_n)) = \bigcup_{k \geq k_0} T_P(I_k)$. So it remains to show that $\bigcup_{k \geq k_0} T_P(I_k) = T_P(\bigcup_{k \geq k_0} I_k)$, which is true since T_P is Scott-continuous. ■

Applying part (1) of the Rutten-Smyth Theorem yields

5.8 Corollary For any definite program P, T_P has a least fixed point $\lim T_P^n(\emptyset)$.

Now we will discuss another way of casting a domain into a quasi-ultrametric space, using similar techniques to those employed in Section 4.1. We follow the approach in [Smy91] and [Sed97].

5.9 Definition Let D be a domain and $r : D_c \to \mathbb{N}$ a rank function such that $r^{-1}(n)$ is a finite set for each $n \in \mathbb{N}$. Define $d_r : D \times D \to \mathbb{R}$ by

$$d_r(x, y) := \inf\{2^{-n} \mid (c \leq x \Longrightarrow c \leq y) \forall c \in D_c \text{ with } r(c) < n\}.$$

Then (D, d_r) is called the quasi-ultrametric space *induced by* r.

It is straightforward to see that (D, d_r) is indeed a quasi-ultrametric space. Note that $d_r^* = \varrho$, where ϱ is the ultrametric from Definition 4.14.

The following notion will be further discussed in Section 6.

5.10 Definition (totally bounded) A gms (X, d) is called *totally bounded* if, for every $\varepsilon > 0$, there exists a finite set $E \subseteq X$ such that for every $y \in X$ there exists an $e \in E$ with $d^*(e, y) < \varepsilon$.

In order to discuss the relationships between quasi-metrics and the atomic topology, we need the following observation:

5.11 Proposition (see [Smy91]) Let (X, d) be a totally bounded gms and (x_n) a CS in X. Then, for all $\varepsilon > 0$, there exists $k \in \mathbb{N}$ such that for all $l, m \geq k$, $d^*(x_l, x_m) < \varepsilon$[13].

Proof: Choose $\varepsilon > 0$ and a finite subset $E \subseteq X$ together with a map $h : \mathbb{N} \to E$ such that $d^*(x_n, h(n)) < \frac{\varepsilon}{3}$, which is possible by total boundedness. Since (x_n) is a CS, there exists $k_0 \in \mathbb{N}$ such that for all $m \geq l \geq k_0$, $d(x_l, x_m) < \frac{\varepsilon}{3}$. Now choose $k_1 \geq k_0$ such that for every $e \in E$, the set $h^{-1}(e) \cap \{n \mid n \geq k_1\}$ is either infinite or empty. Choose now $l, m \geq k_1$ and let $p \geq l$ be minimal such that $h(p) = h(m)$. Then

$$d(x_l, x_m) \leq d(x_l, x_p) + d(x_p, h(p)) + d(h(p), x_m) < 3 \cdot \frac{\varepsilon}{3} = \varepsilon,$$

and by symmetry $d^*(x_l, x_m) < \varepsilon$. ■

5.2 Quasi-metrics and the Atomic Topology

We define totally bounded quasi-ultrametrics on I_P by using level mappings and show that these strongly connect with the atomic topology.

5.12 Definition Let P be a normal logic program and l a finite level mapping for P. As in Definition 4.8, l induces a rank function

$$r : I_c \to \mathbb{N} : r(I) = \max_{A \in I}\{l(A)\},$$

where we set $I_c := (I_P)_c$ to be the set of all finite subsets of B_P. By Definition 5.9, r induces a quasi-ultrametric d_r on I_P, which we will further denote by d.

5.13 Proposition (I_P, d) is a totally bounded quasi-ultrametric space.

Proof: Choose $\varepsilon = 2^{-n}$ and let $E := 2^{\mathcal{L}_{n+1}}$, which is finite since l is finite. For every $I \in I_P$, choose $e = I \cap \mathcal{L}_{n+1} \in E$. Then $d^*(e, I) < \varepsilon$ as is easily verified. ■

We have the following characterization of CSs in I_P.

[13] A sequence with this property is usually called a *bi-Cauchy sequence*.

5.14 Proposition (see [Sed97]) A sequence (I_n) in (I_P, d) is a CS if and only if for every $n \in \mathbb{N}$ there exists $k_n \in \mathbb{N}$ such that for all $l, m \geq k_n$, $I_l \cap \mathcal{L}_n = I_m \cap \mathcal{L}_n$.

Proof: Let (I_n) be a CS in I_P. Choose $n \in \mathbb{N}$ and let $\varepsilon = 2^{-n}$. Since I_P is totally bounded, there exists $k_n \in \mathbb{N}$ such that for all $l, m \geq k_n$, $d^*(I_l, I_m) \leq 2^{-n}$. By definition of d, $I_l \cap \mathcal{L}_n = I_m \cap \mathcal{L}_n$ for all $l, m \geq k_n$ as required. The converse follows, since the above argument reverses. ■

5.15 Corollary (see [Sed97]) Let (I_n) be a sequence in (I_P, d). Then (I_n) is a CS if and only if (I_n) converges in Q to some I. Moreover, $\lim I_n = I$, so (I_P, d) is complete.

Proof: By Proposition 3.2 and the previous proposition, (I_n) is a CS if and only if (I_n) converges in Q to some I. It is easily verified that $\lim I_n = I$ by noting that $I = \{A \in B_P \mid A \in I_n \text{ eventually}\}$. It follows that (I_P, d) is complete. ■

The previous results allow us to characterize CS-continuity in terms of Q.

5.16 Proposition (see [Sed97]) T_P is CS-continuous (for any finite level mapping) if and ony if T_P is continuous in Q.

Proof: Suppose that T_P is CS-continuous and that (I_n) is an arbitrary sequence in I_P which converges in Q to some $I \in I_P$. Then (I_n) is a CS and by Corollary 5.15, $\lim I_n = I$. By CS-coninuity of T_P, we have $\lim T_P(I_n) = T_P(I)$ and again by Corollary 5.15, we have $T_P(I_n) \to T_P(I)$ in Q as required.

Conversely, suppose T_P is continuous in Q and that (I_n) is a CS with $\lim I_n = I$, say. By Corollary 5.15, $I_n \to I$ in Q and by continuity of T_P in Q, we get $T_P(I_n) \to T_P(I)$, which yields $\lim T_P(I_n) = T_P(I)$, again by Corollary 5.15. ■

We close with the observation that non-expansiveness already implies CS-continuity:

5.17 Proposition (see [Sed97]) If T_P is non-expanding (for any level mapping), then T_P is continuous in Q and hence is CS-continuous[14].

Proof: Let T_P be non-expanding and let (I_n) be a CS with $\lim I_n = I$. Since T_P is non-expansive, we get

$$0 \leq d(T_P(I_n), T_P(I)) \leq d(I_n, I) \to 0 \text{ and } 0 \leq d(T_P(I), T_P(I_n)) \leq d(I, I_n) \to 0$$

by total boundedness of I_P. By definition of d and Proposition 5.14, it follows that $T_P(I_n)$ is a CS and by Proposition 3.2 and the above equations, it converges in Q to $T_P(I)$. Hence $\lim T_P(I_n) = T_P(I)$, again by Corollary 5.15. ■

[14]For an arbitrary gms, it is possible that a function is not CS-continuous but non-expanding, see [Rut96].

Summary

The Rutten-Smyth Theorem is a generalization of both the Banach Contraction Mapping Theorem and the Knaster-Tarski Theorem. We have seen that quasi-metrics provide an alternative approach to the semantics of definite programs, using the Rutten-Smyth Theorem. By employing level mappings, similar to the considerations in section 4, we were able to define quasi-metrics on I_P. It was shown that CS-continuity in these spaces is exactly continuity with respect to the atomic topology, and convergence of CS is exactly convergence in Q.

Problems

Problem 8 Is it possible to weaken the assumption of the Rutten-Smyth Theorem that f is non-expansive?

Problem 9 Connect the Rutten-Smyth Theorem with Theorem 4.4by Pries-Crampe and Ribenboim. Is there a common generalization?

Problem 10 Examine the relationships between generalized metric spaces, P-metric spaces and domains.

Problem 11 Try to find a normal logic program P together with a level mapping such that T_P is not monotonic (nor Scott-continuous) but is non-expanding in order to apply the Rutten-Smyth Theorem in non-trivial cases. Is this at all possible?

6 Compactness of Generalized Metric Spaces

We recall some basic notions and results on metric spaces.

6.1 Definition A metric space (X, d) is *totally bounded*, if for every $\varepsilon > 0$ finitely many ε-spheres cover X.

6.2 Proposition A metric space X is totally bounded if and only if each sequence in X has a Cauchy-subsequence.

6.3 Theorem A metric space is compact if and only if it is complete and totally bounded.

We try to generalize some of the above results to generalized metric spaces. The main problem one has to keep in mind is, that for generalized metric spaces, no notion of a natural underlying topology exists.

6.1 Sequential Compactness

Recall, that a topological space X is called *sequentially compact* if every sequence in X has a convergent subsequence.

6.4 Proposition Let (X, d) be a totally bounded gms. Then every sequence in X has a Cauchy-subsequence.

Proof: Let (x_n) be a sequence in X and $B^*_\varepsilon(x) := \{y \in X \mid d^*(x, y) < \varepsilon\}$. We define inductively a subsequence (y_n) of (x_n) as follows:
Let $y_0 := x_0$ and $A_0 = \{x_n \mid n \in \mathbb{N}\}$.
Since X is totally bounded, there exists a 1-sphere B^*_1 such that the set $\{x_n \mid n \in \mathbb{N}\} \cap B_1 = A_0 \cap B_1$ is infinite. Let $A_1 := A_0 \cap B_1$. Now choose $y_1 \in A_1$.
Inductively, let $y_0, \ldots, y_m, A_0 \supseteq A_1 \cdots \supseteq A_m$ be chosen. Since X is totally bounded, there exists a $\frac{1}{m+1}$-sphere B^*_{m+1} such that $A_m \cap B_{m+1}$ is infinite. Let $A_{m+1} := A_m \cap B_{m+1}$ and choose $y_{m+1} \in A_{m+1}$.
Obviously, (y_n) is a Cauchy-subsequence of (x_n). ■

6.5 Remark The sequence (y_n) as constructed in the last proof is a Cauchy-sequence with respect to the metric d^*.

6.6 Proposition Let (X, d) be a totally bounded and CS-complete gms. Then X is sequentially compact.

Proof: Every sequence in X has a Cauchy-subsequence (by total boundedness) which converges (by CS-completeness). ■

Note, that for the above considerations, no topology on the gms X was fixed. A natural restriction on a underlying topology would be, that limits of CSs are topological limits. Note, that in the case of Example 5.2, CSs are closely related to ω-chains and limits of CSs to suprema of ω-chains. In the next section, we investigate sufficient conditions on X such that X is compact.

6.2 Compactness

Recall, that an indexed subset $(x_\lambda)_{\lambda\in\Lambda}$ of a topological space X is called a *net*, if Λ is directed. $(x_\mu)_{\mu\in M}$ is called a *subnet of* $(x_\lambda)_{\lambda\in\Lambda}$, if M is directed and there exists a function $\varphi : M \to \Lambda$ such that

1. $\mu_1 \leq \mu_2 \Longrightarrow \varphi(\mu_1) \leq \varphi(\mu_2)$ and
2. $\forall\lambda \in \Lambda \exists \mu \in M : \lambda \leq \varphi(\mu)$.

Let $(x_\lambda)_{\lambda\in\Lambda}$ be a net and X a set. We say, that $x_\lambda \in X$ *frequently*, if for every $\lambda_0 \in \Lambda$, there exists $\lambda \geq \lambda_0$, such that $x_\lambda \in X$.

6.7 Definition Let (X, d) be a gms. A net (x_μ) in X is a *(forward-) Cauchy-net* (*CN*), if $\forall\varepsilon > 0 \exists \mu_0 \in M \forall \mu_2 \geq \mu_1 \geq \mu_0 \in M : d(x_{\mu_1}, x_{\mu_2}) < \varepsilon$.
We call X *CN-complete* if X inherits a topology, such that every CN converges.

6.8 Proposition Let (X, d) be a totally bounded gms. Then every CN in X has a Cauchy-subnet.

Proof: Let $(x_\lambda)_{\lambda\in\Lambda}$ be a net in X. Note that by total boundedness, $\forall\varepsilon > 0 \exists x \in X$: $x_\lambda \in B^*_\varepsilon(x)$ frequently.
Choose a 1-sphere B^*_1 such that $x_\lambda \in B^*_1$ frequently and let $A_1 := B^*_1$.
Inductively, let A_k be chosen such that $x_\lambda \in A_k$ frequently. By total boundedness, there exists a $\frac{1}{k+1}$-sphere $B^*_{\frac{1}{k+1}}$ such that $x_\lambda \in A_k \cap B^*_{k+1}$ frequently. Let $A_{k+1} := A_k \cap B^*_{\frac{1}{k+1}}$.
By construction, $A_i \supseteq A_{i+1}$ for all $i \in \mathbb{N}$.
Consider $M := \{(\lambda, A_k) \mid x_\lambda \in A_k\}$, ordered by $(\lambda, A_k) \leq (\mu, A_l) :\Longleftrightarrow (\lambda \leq \mu$ and $k \leq l)$. Let $x_{(\lambda,A)} := x_\lambda$.
We show that (1) $(x_{(\lambda,A)})_{(\lambda,A)\in M}$ is a subnet of $(x_\lambda)_{\lambda\in\Lambda}$, and furthermore, that (2) $(x_{(\lambda,A)})_{(\lambda,A)\in M}$ is a CN. Then, the proof is complete.
(1) Let $\varphi : M \to \Lambda : (\lambda, A) \mapsto \lambda$. Obviously, φ has the required properties. So we only have to show that M is directed. To see this, let (λ, A_k), (μ, A_l) be in M, and without any loss of generality, $k \leq l$. From the construction of M it follows that there is $\nu \in \lambda$ with $\lambda, \mu \leq \nu$ and $x_\nu \in A_l$ (note that x_λ is frequently in A_l). Therefore, we have $x_{(\lambda,A_k)}, x_{(\mu,A_l)} \leq x_{(\nu,A_l)}$.
(2) Let $\varepsilon > 0$, and choose $n \in \mathbb{N}$ such that $n \geq \frac{2}{\varepsilon}$. By construction, there exists $x_{(\lambda,A_n)} \in A_n$ and for every $x_{(\mu,B)} \geq x_{(\lambda,A_n)}$, we have $x_{(\mu,B)} \in A_n$. Note that

$A_n \subseteq B^*_{\frac{1}{n}}$ and therefore for all $x, y \in A_n$, we have $d(x,y) < \frac{1}{n}$. Let x_0 be the midpoint of $B^*_{\frac{1}{n}} = B^*_{\frac{1}{n}}(x_0)$. Now let $x_{(\mu_2,B_2)} \geq x_{(\mu_1,B_1)} \geq x_{(\lambda,A_n)}$. Then $d(x_{(\mu_1,B_1)}, x_{(\mu_2,B_2)}) \leq d(x_{(\mu_1,B_1)}, x_0) + d(x_0, x_{(\mu_2,B_2)}) < 2 \cdot \frac{1}{n} = \varepsilon$. ■

6.9 Remark The CN constructed in the proof above is a Cauchy-net with respect to (X, d^*).

6.10 Theorem Let X be a totally bounded and CN-complete gms. Then X is compact.

Proof: Every net in X has a Cauchy-subnet (by total boundedness), which converges (by CN-completeness). ■

Note, that for the above considerations no topology on the gms X was fixed. The only requirement for the underlying topology was the existence of limits of CNs.

Summary

We have seen, that total boundedness and CS-completeness imply sequential compactness, and that total boundedness and CN-completeness imply compactness. The results were obtained by methods very similar to those for metric spaces.

Problems

Problem 12 Investigate the importance of CN-completeness in the context of the relationships between domains and generalized metric spaces, especially, how CNs connect up with directed sets.

Problem 13 Try to characterize the topologies underlying the notions of CN-completeness. Connect this to the topologies in [BBR96], if possible.

Problem 14 Investigate the relationships between the notion of CN-completeness for generalized metric spaces and spherical completeness for P-metric spaces, perhaps by adopting a domain-theoretical point of view.

7 Partitioning Programs

As is easily seen by Program 7, considering orbits of the immediate consequence operator can not be sufficient for finding models for normal logic programs in general. We therefore use another approach here, namely by partitioning the program or its Herbrand base, and applying the respective operators subsequently.

7.1 Stratified Programs

We begin by briefly reviewing stratified logic programs as discussed in [ABW88].

7.1 Definition (stratified program) A program P is called *stratified* if there is a partition $P = P_1 \dot\cup \ldots \dot\cup P_m$ of P into subprograms such that the following conditions hold for $i = 1, \ldots, m$:

(1) If a predicate symbol occurs in a positive literal in P_i, then it occurs only in heads of clauses contained in $\bigcup_{j \leq i} P_j$.

(2) If a predicate symbol occurs in a negative literal in P_i, then it occurs only in heads of clauses contained in $\bigcup_{j < i} P_j$.

The set P_1 may be empty. Each P_i is called a *stratum* of P.

Note that P_1 can always be chosen to be definite[15].

We now define an operator on stratified programs.

7.2 Definition Let $P = P_1 \dot\cup \ldots \dot\cup P_m$ be a stratified program and $I \in I_P$. We define recursively

$$\begin{aligned} T_P \uparrow 0(I) &:= I, \\ T_P \uparrow (n+1)(I) &:= T_P(T_P \uparrow n(I)) \cup T_P \uparrow n(I) \text{ for } n \geq 0 \text{ and} \\ T_P \uparrow \omega(I) &:= \bigcup_{n=0}^{\infty} T_P \uparrow n(I). \end{aligned}$$

The following theorem is given in [ABW88].

7.3 Theorem Let $P = P_1 \dot\cup \ldots \dot\cup P_m$ be a stratified program and define M_i recursively by

$$\begin{aligned} M_0 &:= \emptyset, \text{ and} \\ M_i &:= T_{P_i} \uparrow \omega(M_{i-1}), \end{aligned}$$

for $i = 1, \ldots, m$. Finally, set $M_P := M_m$. Then M_P is a minimal supported model for P.

[15] We consider an empty program to be definite.

We do not give the proof of this theorem here since the results of Section 7.2 below generalize it (see Theorem 7.24). The main advantage of the approach described above is that it gives control over negation in a way displayed by the following general lemma, which is the heart of the proof of Theorem 7.3 (see [ABW88, Lemma 10]). We will also need it later on.

7.4 Lemma (see [ABW88, Lemma 10] Let P' be an arbitrary normal logic program and P a subprogram of P' in the sense that every clause in P is a clause in P'. Denote the subset of B_P consisting of all the atoms which occur negated in clauses in ground(P) by $\mathcal{N}_P$.

If $I \subseteq J \in I_{P'}$ and $I \cap \mathcal{N}_P = J \cap \mathcal{N}_P$, then $T_P(I) \subseteq T_P(J)$. In particular, if $P = P_1 \dot\cup \ldots, \dot\cup P_m$ is stratified, then $T_{P_i} \uparrow (k+1)(M_{i-1}) = T_{P_i}(T_{P_i} \uparrow k(M_{i-1})) \cup M_{i-1}$ for $i = 1, \ldots, m$ and $k \in \mathbb{N}$.

Proof: Let $A \in T_P(I)$. Then there is a clause $A \leftarrow A_1, \ldots, A_{l_1}, \neg B_1, \ldots, \neg B_{l_1}$ in ground(P) such that $A_k \in I$ and $B_l \notin I$ for $k = 1, \ldots, k_1$ and $l = 1, \ldots, l_1$. Since $I \subseteq J$, $A_k \in J$ for all k, and since $B_l \in \mathcal{N}_P$ for all l, $B_l \notin J$. Hence $A \in T_P(J)$, which proves the first statement. For the second, note that the statement trivially holds for $k = 0$. Suppose now that it holds for some $k \geq 0$. Then $T_{P_i} \uparrow (k+2)(M_{i-1}) = T_{P_i}(T_{P_i} \uparrow (k+1)(M_{i-1})) \cup T_{P_i} \uparrow (k+1)(M_{i-1}) = T_{P_i}(T_{P_i} \uparrow (k+1)(M_{i-1})) \cup T_{P_i}(T_{P_i} \uparrow k(M_{i-1})) \cup M_{i-1}$. Since $T_{P_i} \uparrow (k+1)(M_{i-1}) \supseteq T_{P_i} \uparrow k(M_{i-1})$, it follows by the first statement that $T_{P_i}(T_{P_i} \uparrow (k+1)(M_{i-1})) \supseteq T_{P_i}(T_{P_i} \uparrow k(M_{i-1}))$, which finishes the proof. ■

Informally, the lemma says that T_P is monotone as long as its arguments do not differ on the elements of $\mathcal{N}_P$.

We now give an alternative characterization of the model M_P for stratified P, as defined above.

7.5 Definition For a given normal logic program P and $A \in B_P$ the *resolution tree* $R_P(A)$ *for* A (*with respect to* P) is defined recursively as follows:

A is the root of $R_P(A)$.

Let

$$N = A_1, \ldots, A_{n_1}, \neg B_1, \ldots, \neg B_{n_2}$$

be a node in $R_P(A)$. Then for every $i = 1, \ldots, n_1$ and every

$$A_i \leftarrow C_1, \ldots, C_{m_1}, \neg D_1, \ldots, \neg D_{m_2} \in \text{ground}(P),$$

$$A_1, \ldots, A_{i-1}, A_{i+1}, \ldots, A_{n_1}, C_1, \ldots, C_{m_1}, \neg B_1, \ldots, \neg B_{n_2}, \neg D_1, \ldots, \neg D_{m_2}$$

is a daughter of N. A branch with an empty leaf is called a *success branch* of A. A branch with a leaf such that all of its literals are negative is called a *negative branch* of A.

In the following, let $P = P_1 \dot\cup \ldots \dot\cup P_m$ be a stratified logic program with strata $P_1, \ldots, P_m$. Furthermore, let M_i for $i = 1, \ldots, m$ be defined as in Theorem 7.3, let M_P

be the minimal supported model of P as given there and let T_i denote T_{P_i}. Since it was shown in [ABW88] that M_P is independent of the stratification of P, we can assume without loss of generality that each predicate in P is defined in exactly one P_i for some $i \in \{1, \ldots, m\}$ and P_1 is definite or empty.

7.6 Lemma Let P_1 be a definite program and let $P = P_1 \dot{\cup} \ldots \dot{\cup} P_m$ be a stratification of P. Then the following hold for $i = 1, \ldots, m-1$:

1. $A \in T^n_{P_1}(\emptyset)$ eventually if and only if $R_{P_1}(A)$ has a success branch.
2. $A \in T_{P_{i+1}} \uparrow n(M_i)$ eventually if and only if $R_{P_i}(A)$ has either a success branch or a branch with leaf $A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ such that $A_k \in M_i$ and $B_l \notin M_i$ for $k = 1, \ldots, k_1$ and $l = 1, \ldots, l_1$.

Proof: The first statement follows from the definitions and the fact that T_{P_1} is monotonic. The second statement follows from the definitions and Lemma 7.4. ■

7.7 Remark Suppose $A \in B_P$ matches some head in P_i. Then all positive literals occuring in $R_P(A)$ are defined in some P_j with $j \leq i$ and all negative literals occuring in $R_P(A)$ are defined in some P_k with $k < i$.

If P is semi-strictly level-decreasing with respect to an ω-level mapping and $A \in B_P$ with $l(A) = n$, then every atom in $R_P(A)$ is of level $\leq n$ and every negated atom in $R_P(A)$ is of level $< n$. Analogous conditions hold for (strictly) level-decreasing programs.

7.8 Definition We define the set $J_P = \bigcup_{i=0}^{m} J_i \subseteq B_P$ recursively as follows:

J_1 is the set of all ground instances A of predicates matching some head in P_1 such that $R_{P_1}(A)$ has a success branch.

J_{i+1} is J_i unified with the set of all ground instances A of predicates matching some head in P_{i+1} such that $R_{P_{i+1}}(A)$ has a success branch or a branch with leaf $A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_2}$, where $A_k \in J_i$ and $B_l \notin J_i$ for $k = 1, \ldots, k_1$ and $l = 1, \ldots, l_1$. Since P is stratified, every J_i, and hence J_P, is well-defined.

7.9 Lemma For $i = 1, \ldots, m$, we have $J_i = M_i$.

Proof: Recall that P_1 is definite, so T_1 is monotone. It follows that $T_1^k(\emptyset) = T_1 \uparrow k(\emptyset)$.

Let $A \in J_1$. By definition of a stratification, every literal occuring in $R_{P_1}(A)$ is positive. By definition of J_1, A has a success branch in $R_{P_1}(A)$. By Lemma 7.6, $A \in T_1^k(\emptyset)$ for some $k \in \mathbb{N}$ and therefore $A \in T_1 \uparrow \omega(\emptyset) = M_1$.

Let $A \in M_1 = T_1 \uparrow \omega(\emptyset)$. Since P_1 is definite, there exists some $n_0 \in \mathbb{N}$ such that $A \in T_1^n(\emptyset)$ for all $n \geq n_0$. Hence, $R_{P_1}(A)$ has a success branch, and it follows that $A \in J_1$.

We proceed by induction on $i = 1, \ldots, m-1$

Let $A \in J_{i+1}$ for some $i > 0$. Then $R_{P_{i+1}}(A)$ has a leaf $A_1, \ldots, A_{n_1}, B_1, \ldots, \neg B_{n_2}$ such that for all $k = 1, \ldots, n_1$ and for all $l = 1, \ldots, n_2$, $A_{k_1} \in J_i$ and $B_{l_1} \notin J_i$. By the induction hypothesis, $J_i = M_i$, and from Lemma 7.6 it follows that for some $n_0 \in \mathbb{N}$ we have $A \in T_{i+1} \uparrow n_0(M_i) \subseteq M_{i+1}$.

Let $A \in M_{i+1}$ for some $i > 0$. Then there exists some $n_0 \in \mathbb{N}$ such that $A \in T_{i+1} \uparrow n(M_i)$ for all $n \geq n_0$. By Lemma 7.6, $R_{P_1}(A)$ has a leaf $A_1, \ldots, A_{n_1}, B_1, \ldots, \neg B_{n_2}$ such that for all $k = 1, \ldots, n_1$ and for all $l = 1, \ldots, n_2$, $A_{k_1} \in J_i$ and $B_{l_1} \notin J_i$. By the induction hypothesis, $J_i = M_i$, and by definition of J_{i+1} it follows that $A \in J_{i+1}$. ■

The following theorem is new.

7.10 Theorem For every stratified logic program P, $J_P = M_P$.

Proof: By Lemma 7.9 we have $J_P = \bigcup_{i=1}^{m} J_i = \bigcup_{i=1}^{m} M_i = M_P$. ■

The remainder of this section is devoted to the study of stratified programs and of programs which are both stratified and semi-strictly level-decreasing with respect to a finite level mapping.

7.11 Definition Let P be a normal logic program with a finite level mapping l as defined in 4.8. We define the *finitely at level n intersected immediate consequence map*[16] $T_P^{(n)} : \mathbf{2}^{\mathcal{L}_{n+1}} \to \mathbf{2}^{\mathcal{L}_{n+1}}$ for a (normal) logic program P by $T_P^{(n)}(I) := T_P(I) \cap \mathcal{L}_{n+1}$. Furthermore, for $T : I_P \to I_P$ let

$$T \Uparrow 0(I) := I$$

and inductively

$$T \Uparrow (k+1)(I) := T(T \Uparrow k(I)) \cup I \text{ for } k \geq 0.$$

7.12 Lemma For stratified P and for every $k \in \mathbb{N}$, $T_i \Uparrow k(M_{i-1}) = T_i \uparrow k(M_{i-1})$.

Proof: The statement obviously holds for $k = 0$ and every i. We proceed by induction on k. Suppose, the statement holds for $k \geq 0$. Then by Lemma 7.4, $T_i \Uparrow (k+1)(M_{i-1}) = T_i(T_i \Uparrow k(M_{i-1})) \cup M_{i-1} = T_i(T_i \uparrow k(M_{i-1})) \cup M_{i-1} = T_i(T_i \uparrow k(M_{i-1})) \cup T_i \uparrow k(M_{i-1}) = T_i \uparrow (k+1)(M_{i-1})$. ■

7.13 Definition Let P be a logic program which is semi-strictly level-decreasing with respect to a finite level mapping, and which is stratified. We define the sequence $(I_n) \in I_P$ inductively as follows:

Since P_1 is definite and $\mathcal{L}_n$ is finite for every n, the sequence $(T_1^{(n)} \Uparrow k(\emptyset))_{k \in \mathbb{N}}$ is eventually constant, say equal to $I_{n,1}$.

By Lemma 7.12 and 7.4, $(T_{i+1} \Uparrow k(I_{n,i}))_{k \in \mathbb{N}}$ is increasing with k and hence is eventually constant, say equal to $I_{n,i}$.

Now define $I_n := I_{n,m}$.

[16] In [Hit97], a Prolog program was developped which computes the iterates of the operator $T^{(n)}$ in the case of a finite level mapping.

7.14 Example For Program 10, we define a level mapping $l : B_P \to B_P$ by setting $l(A)$ to be equal to the number of function symbols s occuring in A. We then get $I_{0,1} = \{q(0)\}$, $I_0 = I_{0,2} = \{q(0)\}$ and $I_n = \{q(0)\}$ for all $n \in \mathbb{N}$.

The following result is new.

7.15 Theorem Let P be a logic program which is semi-strictly level-decreasing with respect to a finite level mapping l, and which is stratified. Then the sequence $(I_n)_{n\in\mathbb{N}}$ as defined above converges in Q to M_P.

Proof: We show $I_n \to J_P$ in Q.

(i) $A \in B_{P_1}$: Let $A \in J_1$ with $l(A) = n_0$. Since P_1 is definite, $R_{P_1}(A)$ has a success branch such that all atoms occuring in any node of this branch are of level $\leq n_0$. By Lemma 7.6, $A \in T_1^{(n)} \Uparrow k$ for every $n \geq n_0$ and every $k \geq k_0$ for some $k_0 \in \mathbb{N}$. Hence, $A \in I_n$ eventually.

Let $A \notin J_1$. Then $R_{P_1}(A)$ has no success branch and therefore, for every $n \in \mathbb{N}$, $A \notin I_n$. We proceed by induction on $i = 1, \ldots, m-1$.

(ii) $A \in B_{P_{i+1}}$ matches some head in ground(P_{i+1}): Let $A \in J_{i+1}$ with $l(A) = n_0$. If $R_{P_{i+1}}$ has a success branch, then by the same argument as for J_1, $A \in I_n$ eventually. If $R_{P_{i+1}}(A)$ has no success branch, then it has a branch with leaf $A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ such that $A_k \in J_i$ and $B_l \notin J_i$ for all $k = 1, \ldots, k_1$ and $l = 1, \ldots, l_1$. By the induction hypothesis, $A_k \in I_n$ and $B_l \notin I_n$ eventually, say for all $n \geq n_1$. It follows that for all $n \geq n_1$, we have $A_k \in I_{n,i}$ and $B_l \notin I_{n,i}$, and therefore $A \in I_{n,i+1} \subseteq I_n$ eventually.

Let $A \notin J_{i+1}$. Then $R_{P_{i+1}}(A)$ does not have a success branch and does not have a leaf $A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ with $A_k \in J_i$ and $B_l \notin J_i$ for all $k = 1, \ldots, k_1$ and $l = 1, \ldots, l_1$. By the induction hypothesis, there is either some $A_{k_0} \notin I_n$ eventually or some $B_{l_0} \in I_n$ eventually. In either case, $A \notin I_n$ eventually. ∎

7.2 Semi-strictly Level-decreasing Progams

We study the class of programs which are semi-strictly level-decreasing with respect to an ω-level mapping[17] and construct a minimal supported model for them which coincides with M_P if P is stratified. In fact, the programs considered here are all *locally stratified*, as defined in [Prz88], where it was shown that these programs have a unique perfect model[18] (as defined there). The methods used here are quite different from those employed in the cited paper of Przymusinski.

For the remainder of this section, all level mappings are considered to be ω-level mappings.

7.16 Definition (Finitely restricted immediate consequence map) Let P be a normal logic program with a level mapping l. Let $P_{[n]} :=$ ground(P, n) be the set of all

[17] In [SH97c], we extended the approach to arbitrary level-mappings.

[18] It should be noted that in [SH97c] we have shown that the model constructed in this section is exactly the unique perfect model obtained in [Prz88].

clauses in ground(P) in which only atoms of level $\leq n$ occur. We define the *finitely up to level n restricted immediate consequence map*[19] $T_{[n]} : \mathbf{2}^{\mathcal{L}_{n+1}} \to \mathbf{2}^{\mathcal{L}_{n+1}}$ by $T_{[n]}(I) := T_{P_{[n]}}(I)$.

The following proposition is immediate:

7.17 Proposition Let P be a level-decreasing logic program and let $I \subseteq B_P$. Then $T_{[n]}(I) = T^{(n)}(I)$ for every $n \in \mathbb{N}$.

We now construct an interpretation by using the finitely intersected immediate consequence map for semi-strictly level-decreasing programs. This interpretation will later be shown to be a minimal supported model for the program in question.

7.18 Definition Let P be a semi-strictly level-decreasing logic program. We define the sequence $(I_n)_{n\in\mathbb{N}}$ inductively as follows:

For each $m \in \mathbb{N}$, let $I_{[0,m]} := T^m_{[0]}(\emptyset)$, and set $I_0 := \bigcup_{m=0}^{\infty} I_{[0,m]}$.

Furthermore, for each $m \in \mathbb{N}$ and $n > 0$, let $I_{[n+1,m]} := T^m_{[n+1]}(I_n)$ and set $I_{n+1} := \bigcup_{m=0}^{\infty} I_{[n+1,m]}$.

Finally, let $I_{[P]} := \bigcup_{n=0}^{\infty} I_n$.

The main technical lemma needed for showing that $I_{[P]}$ is indeed a minimal supported model is the following.

7.19 Lemma (see [SH97c]) Let P be a normal logic program which is semi-strictly level-decreasing with respect to an ω-level mapping l. Then the following statements hold.

1. For every n, the sequence $(I_{[n,m]})$ is monotonic increasing in m.
2. For every n, I_n is a fixed point of $T_{[n]}$.
3. The sequence (I_n) is monotonic increasing.
4. If $l(B) \leq n$ and $B \notin I_n$, where $B \in B_P$, then for every $r \in N$, we have $B \notin I_{n+r}$ and hence $B \notin I_{[P]}$. In particular, if $l(B) \leq n$ and $B \notin I_{[n+1,m]}$ for some m, then $B \notin I_n$ and hence $B \notin I_{[P]}$.

Proof: We begin by noting the technical fact that, for each $n \in N$, we can partition $P_{[n+1]}$ as $P_{[n]} \cup P(n+1)$, where $P(n+1)$ denotes the subset of ground(P) consisting of those clauses whose head has level $n+1$. Thus, $T_{[n+1]}(I) = T_{[n]}(I) \cup T_{P(n+1)}(I)$ for any $I \in I_P$; note that if $A \in T_{P(n+1)}(I)$, then $l(A) = n+1$.

It will be convenient to present the proof, which is by induction, in a sequence of steps, and our induction hypothesis is the proposition: "For some n_0 and each n satisfying

[19] In [Hit97], I developped a Prolog program which computes the iterates of the operator $T_{[n]}$ for a finite level mapping.

$0 \leq n \leq n_0$, the sequence $(I_{[n,m]})$ is monotonic increasing in m and I_n is a fixed point of $T_{[n]}$".

Step 1. The proposition just stated is true of course for $n_0 = 0$ since $P_{[0]}$ is a definite program and I_0 is the least fixed point of $T_{[0]}$ by the classical construction. Suppose the proposition true for some $n_0 > 0$. Then in particular it holds with $n = n_0$ and hence I_{n_0} is a fixed point of $T_{[n_0]}$. We show the proposition true for $n_0 + 1$ and hence that it is true for all n_0. It will ease notation to write k for n_0 in what follows.

Step 2. We establish the recursion equations:

$$I_{[k+1,0]} = I_k \text{ and}$$
$$I_{[k+1,m+1]} = I_k \cup T_{P(k+1)}(I_{[k+1,m]}) \text{ for } m \geq 0$$

and the first is immediate. Putting $m = 0$, we have $I_{[k+1,1]} = T_{[k+1]}(I_k) = T_{[k]}(I_k) \cup T_{P(k+1)}(I_k) = I_k \cup T_{P(k+1)}(I_k) = I_k \cup T_{P(k+1)}(I_{[k+1,0]})$, using the fact that I_k is a fixed point of $T_{[k]}$. Now suppose that the second of these equations holds for some $m > 0$. Then $I_{[k+1,(m+1)+1]} = T_{[k+1]}(I_{[k+1,m+1]}) = T_{[k]}(I_{[k+1,m+1]}) \cup T_{P(k+1)}(I_{[k+1,m+1]}) = T_{[k]}(I_k \cup T_{P(k+1)}(I_{[k+1,m]})) \cup T_{P(k+1)}(I_{[k+1,m+1]})$. It suffices, therefore, to show that $T_{[k]}(I_k \cup T_{P(k+1)}(I_{[k+1,m]})) = I_k$. Suppose that $A \in T_{[k]}(I_k \cup T_{P(k+1)}(I_{[k+1,m]}))$. Then there is a clause $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ in $P_{[k]}$ such that $A_1, \ldots, A_{k_1} \in I_k \cup T_{P(k+1)}(I_{[k+1,m]})$ and $B_1, \ldots, B_{l_1} \notin I_k \cup T_{P(k+1)}(I_{[k+1,m]})$. But then level considerations and the hypothesis concerning P imply that $A_1, \ldots, A_{k_1} \in I_k$ and $B_1, \ldots, B_{l_1} \notin I_k$. Therefore, $A \in T_{[k]}(I_k) = I_k$ and we have the inclusion $T_{[k]}(I_k \cup T_{P(k+1)}(I_{[k+1,m]})) \subseteq I_k$. The reverse inclusion is demonstrated in like fashion showing that the second of the recursion equations holds with m replaced by $m + 1$ and hence, by induction, that it holds for all m.

Step 3. We have the inclusions $T_{P(k+1)}(I_k) \subseteq T_{P(k+1)}(I_k \cup T_{P(k+1)}(I_k)) \subseteq T_{P(k+1)}(I_k \cup T_{P(k+1)}(I_k \cup T_{P(k+1)}(I_k))) \ldots$. These inclusions are established by methods similar to those we have just employed and we omit the details.

It is now clear from this fact and the recursion equations in Step 2 that $(I_{[k+1,m]})$ is monotonic increasing in m. Since monotonic increasing sequences converge to their union in Q, see Proposition 3.2, and $I_{[k+1,m]}$ is an iterate of I_k, it now follows from Theorem 3.6 that I_{k+1} is a model for $P_{[k+1]}$.

Step 4. If $B \in B_P$ and $l(B) \leq k$, then $B \in I_{k+1}$ iff $B \in I_k$ (where k still denotes n_0). Indeed, if $B \in I_k$, then it is clear from the recursion equations of Step 2 that $B \in I_{k+1}$. On the other hand, if $B \notin I_k$, then it is equally clear from the recursion equations and level considerations that, for every m, $B \notin I_{[k+1,m]}$ and hence that $B \notin I_{k+1}$, as required.

Step 5. One of the key technical features of our construction is the control it gives over negation, and we illustrate this observation by showing next that I_{k+1} is a supported model for $P_{[k+1]}$. To do this, suppose that $A \in I_{k+1} = \bigcup_{m=0}^{\infty} I_{[k+1,m]}$. Then there is $m_0 \in \mathbb{N}$ such that $A \in I_{[k+1,m+1]} = T_{[k+1]}^{m+1}(I_k)$ for all $m \geq m_0$. Thus, $A \in T_{[k+1]}(T_{[k+1]}^{m_0}(I_k)) = T_{[k+1]}(I_{[k+1,m_0]})$. Hence, there is a clause $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ in $P_{[k+1]}$ such that each $A_i \in I_{[k+1,m_0]}$ and no $B_j \in I_{[k+1,m_0]}$. But $l(B_j) \leq k$ for each j since P is semi-strictly level-decreasing. Since $B_j \notin I_{[k+1,m_0]}$, we now see from the recursion equations that $B_j \notin I_k$. From the result in Step 4 we now deduce that, for each j, $B_j \notin I_{k+1}$. Since it is obvious that each A_i belongs to I_{k+1}, we obtain that $A \in T_{[k+1]}(I_{k+1})$. Thus, $I_{k+1} \subseteq T_{[k+1]}(I_{k+1})$ and therefore I_{k+1} is a supported model for $P_{[k+1]}$, or a fixed point of $T_{[k+1]}$, as required.

Thus, the induction hypothesis holds for $n_0 + 1$ showing therefore that it holds for all n_0. Thus, for every n, $I_{[n,m]}$ is monotonic increasing in m and I_n is a fixed point of $T_{[n]}$, establishing statements 1 and 2. It is immediate from the construction that the sequence (I_n) is monotonic increasing, which establishes statement 3. As far as statement 4 is concerned, it follows (by iterating the result of Step 4) that if $l(B) \leq n$ and $B \notin I_n$, then, for every $r \in \mathbb{N}$, we have that $B \notin I_{n+r}$ and hence that $B \notin I_{[P]}$. The final statement in 4 results from the argument given in the previous paragraph showing that $B_j \notin I_k$. ■

Note that it now follows that the recursion equations established in the previous proof hold for all $k \in \mathbb{N}$.

7.20 Remark Let P be a strictly level-decreasing program with respect to an ω-level mapping. It is easy to see that for all $n, m \in \mathbb{N}$, $I_{[n+1,m]} = I_n \cup T_{P(n+1)}(I_n)$, so that the iterates become constant after one step.

7.21 Lemma (see [SH97c]) $I := I_{[P]}$ is a supported model for P.

Proof: We show first that $T_P(I) \subseteq I$. Let $A \in T_P(I)$. Then there is a ground instance $A \leftarrow A_1, \dots, A_{k_1}, \neg B_1, \dots, \neg B_{l_1}$ of a clause in P such that $A_1, \dots, A_{k_1} \in I$ and $B_1, \dots, B_{l_1} \notin I$. By monotonicity of $(I_n)_{n\in\mathbb{N}}$, the fact that $I = \bigcup I_n$ and part 4 of Lemma 7.19 there is $n_0 \in \mathbb{N}$ such that, for all $n \geq n_0$, $A_1, \dots, A_{k_1} \in I_n$ and $B_1, \dots, B_{l_1} \notin I_n$. By definition, $A \in T_{[\max\{n_0, l(A)\}]}(I_{n_0}) \subseteq I$ as required. So I is a model for P.

It remains to show that $T_P(I) \supseteq I$. Let $A \in I$. By monotonicity of $(I_n)_{n\in\mathbb{N}}$, there is $n_0 \in \mathbb{N}$ such that $A \in I_n$ for all $n \geq n_0$. Thus, there is $m_0 \in \mathbb{N}$ such that $A \in I_{[n_0,m]}$ for all $m > m_0$. By definition of $I_{[n_0,m]}$, there is a clause $A \leftarrow A_1, \dots, A_{k_1}, \neg B_1, \dots, \neg B_{l_1}$ in $P_{[n_0]}$ such that $A_1, \dots, A_{k_1} \in I_{[n_0,m_0]} \subseteq I$ and $B_1, \dots, B_{l_1} \notin I_{[n_0,m_0]}$. Since $l(B_j) < n_0$ for all $j = 1, \dots, l_1$ and by part 4 of Lemma 7.19, $B_1, \dots, B_{l_1} \notin I$. By virtue of the clause $A \leftarrow A_1, \dots, A_{k_1}, \neg B_1, \dots, \neg B_{l_1}$, it follows that $A \in T_P(I)$ as required. ■

The obtained model $I_{[P]}$ is not only supported, but is also minimal:

7.22 Theorem (see [SH97c]) Let P be a semi-strictly level decreasing logic program and let $I := I_{[P]}$ be defined as in Definition 7.18. Then $I_{[P]}$ is a supported and minimal model for P.

Proof: By the previous lemma, it remains to show that I is a minimal model. We show this by induction in establishing the following proposition: "If $J \subseteq I$ and $T_P(J) \subseteq J$, then $I_{[n,m]} \subseteq J$ for all $m, n \in \mathbb{N}$."

Now $I_0 = \bigcup I_{[0,m]} \subseteq J$ since $P_{[0]}$ is definite and I_0 is the least model of $P_{[0]}$; thus, the proposition holds for $n = 0$.

Assume now that the proposition holds for some $n_0 > 0$ and all $m \in \mathbb{N}$, so $I_{[n_0,m]} \subseteq J$ for all m. We show by induction on m that $I_{[n_0+1,m]} \subseteq J$ for all m. Indeed, with $m = 0$ we have $I_{[n_0+1,0]} = I_{n_0} = \bigcup_{m=0}^{\infty} I_{[n_0,m]} \subseteq J$ by our current assumption. Suppose, therefore, that $I_{[n_0+1,m_0]} \subseteq J$ for some $m_0 > 0$. Let $A \in I_{[n_0+1,m_0+1]} = T_{[n_0+1]}(T^{m_0}_{[n_0+1]}(I_{n_0}))$. Then there is a clause $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ in $P_{[n_0+1]}$ such that $A_1, \ldots, A_{k_1} \in T^{m_0}_{[n_0+1]}(I_{n_0}) = I_{[n_0+1,m_0]}$ and $B_1, \ldots, B_{l_1} \notin I_{[n_0+1,m_0]}$. But $l(B_j) \leq n_0$ for each j. Applying Lemma 7.19 part 4 again we see that no B_j belongs to $I_{[P]}$ and consequently no B_j belongs to J since $J \subseteq I_{[P]}$. Since $I_{[n_0+1,m_0]} \subseteq J$ by assumption, we have $A_1, \ldots, A_{k_1} \in J$. Therefore, $A \in T_{[n_0+1]}(J) \subseteq T_P(J) \subseteq J$, from which we obtain that $I_{[n_0+1,m_0+1]} \subseteq J$ as required. ■

For an application, see Program 9.

The model obtained thus coincides with M_P if P is stratified.

7.23 Proposition Every stratified logic program is semi-strictly level-decreasing.

Proof: Let $P = P_1 \dot\cup \ldots \dot\cup P_m$ be stratified as in Definition 7.1. We define a level mapping l by $l(A) := i$ if A is an element of B_P in which the predicate symbol occuring is defined in P_{i+1}, and set $l(A) = 0$ for every $A \in B_P$, which is not defined in P. ■

7.24 Theorem Let $P = P_1 \dot\cup \ldots \dot\cup P_m$ be a stratified logic program with level mapping as defined in the proof of Proposition 7.23 and let $I_{[P]}$ be defined as in Definition 7.18. Then $I_{[P]} = M_P$.

Proof: We will show by induction that $I_k = M_{k+1}$ for $k = 0, 1, \ldots, m-1$ and that $I_k = M_m$ for $k \geq m$. From this we clearly have $I_{[P]} = M_m = M_P$ as required.

Now $P_{[0]} = \text{ground}(P_1)$ is definite, even if empty, and so it is immediate that $I_0 = M_1$. So suppose next that $I_k = M_{k+1}$ for some $k > 0$. Then $I_{[k+1,0]} = I_k = M_{k+1} = T_{P_{k+2}} \uparrow 0(M_{k+1})$. So now suppose that $I_{[k+1,m]} = T_{P_{k+2}} \uparrow m(M_{k+1})$ for some $m > 0$. We have $I_{[k+1,m+1]} = I_k \cup T_{P(k+1)}(I_{[k+1,m]}) = M_{k+1} \cup T_{P_{k+2}}(T_{P_{k+2}} \uparrow m(M_{k+1})) = M_{k+1} \cup T_{P_{k+2}}(T_{P_{k+2}} \Uparrow m(M_{k+1})) = T_{P_{k+2}} \Uparrow (m+1)(M_{k+1}) = T_{P_{k+2}} \uparrow (m+1)(M_{k+1})$, by Lemma 7.12. Therefore, $I_{[k+1,m+1]} = T_{P_{k+2}} \uparrow (m+1)(M_{k+1})$. From this we obtain, by induction, the equality $I_{[k+1,m]} = T_{P_{k+2}} \uparrow m(M_{k+1})$ for all m and with it the equality $I_{k+1} = M_{k+2}$ as required. ■

We close with a comparision of the complexities of the different approaches discussed for strictly and semi-strictly level-decreasing programs:

1. For strictly level-decreasing programs with respect to ω-level mappings, it suffices, as shown in Section 4, to compute the sequence $(T^n_P(\emptyset))$ to obtain the unique supported model for the program, and therefore only a single limit is involved.

2. The approach in Section 7.1 for programs which are stratified and semi-strictly level-decreasing with respect to a finite level mapping requires one to compute the

single sequence (I_n). Moreover, each member of this sequence is itself obtained by a finite computation. Again, therefore, only a single limit is required in this case.

3. The approach of Apt, Blair and Walker [ABW88] or the use of the approach from Section 7.2 in the case of stratified programs requires the computation of the limits of finitely many sequences $(T_{P_{k+1}} \uparrow n(M_k))$.

4. Using the construction in Section 7.2 for semi-strictly level-decreasing programs with respect to an ω-level mapping involves the computation of the limit of the sequence (I_n). Here each I_n is itself obtained by constructing the sequence $(I_{[n,m]})_m$ and its limit. So in this case at most countably many limits have to be computed. If the program is semi-strictly level-decreasing with respect to a finite level mapping, then the sequence $(I_{[n,m]})_m$ stabilises after finitely many steps, and therefore only a single limit needs to be computed.

Summary

We have shown how to obtain minimal supported models for a logic program P by partitioning it or its Herbrand base, and applying the respective immediate consequence operators subsequently. In particular, we constructed minimal supported models for semi-strictly level-decreasing logic programs with ω-level mappings. Additionally, we have given an alternative characterization of the minimal supported standard model M_P for stratified programs.

Problems

Problem 15 In [ABW88], an interpreter for stratified logic programs (without function symbols) was given. Does there exist an interpreter for semi-strictly level-decreasing logic programs P (including function symbols) which derives exactly $I_{[P]}$? See Program 13.

Problem 16 Can the methods from Section 4 be applied to semi-strictly level-decreasing programs by using Theorem 4.4?

Problem 17 Is it possible to extend any of the approaches discussed so far to "level-increasing programs"? See Program 14.

8 Conclusions

We have seen that in order to find models for a given logic program P, it is worth endowing the space I_P of all Herbrand interpretations for P with topologies or topology-like structures. All of those structures discussed in this thesis have a strong relationship with the natural order on I_P, which is set-inclusion, and all of those can be applied to normal logic programs to some extent:

The Scott topology, which we were able to discribe as a natural topology on I_P, determines the semantics of definite programs. The atomic topology Q allows one to find supported models under some semi-syntactic conditions which are all closely related to the absence of local variables. Metric and P-metric spaces were used to find the unique supported model for strictly level-decreasing programs. Approaching semantics by partitioning programs is only indirectly connected with topology, though the atomic topology was used to prove the main theorem 7.22 of Section 7, yielding a minimal supported model for programs which are semi-strictly level-decreasing with respect to ω-level mappings.

There is a number of strong relationships between the sections. The definition of the Scott topology on I_P, given as the positive atomic topology Q^+, and the definition of the Cantor topology on I_P, given as the atomic topology Q, are closely related and this gives a connection between Sections 2 and 3. Section 5 on quasi-metrics relates to Section 2 by giving an alternative approach to definite programs, and to Section 3 by displaying the strong relationships between quasi-metrics obtained from level-mappings and the atomic topology. We have seen in Sections 4 and 5 that domains can be viewed as ultrametric, P-metric and quasi-ultrametric spaces. The relationships between these have still to be investigated, incorporating the results from Section 6. Section 7 connects with Section 3 by employing the atomic topology, and with Section 4 by the closely related notions of strictly and semi-strictly level-decreasing programs.

We raised 17 problems, theoretical and practical in nature, which are worth following in order to further enlighten the relationships between topology and logic programming semantics.

Appendix

A Programs

The following programs are given in Edinburgh-Syntax, as implemented e.g. in SICStus- or SWI-Prolog (cf. [Wie96]). Throughout, P will always denote the actual program inside the examples, and T abbreviates T_P.

Program 1

```
p(0,X,X).
p(s(X),Y,s(Z)) :- p(X,Y,Z).
```

We compute the least Herbrand model of the above definite program by using the Knaster-Tarski Theorem 2.3 and the observations from Section 2. If we denote $T^n(\emptyset)$ by I_n, we get

$$
\begin{aligned}
&I_0\emptyset\\
&I_1T(I_0) = \{p(0, s^k(0), s^k(0)) \mid k \in \mathbb{N}\}\\
&I_2T(I_1) = I_0 \cup \{p(s(0), s^k(0), s^{k+1}(0) \mid k \in \mathbb{N}\}\\
&I_nT(I_{n-1}) = I_{n-1} \cup \{p(s^{n-1}(0), s^k(0), s^{k+n-1}(0) \mid k \in \mathbb{N}\}
\end{aligned}
$$

which yields $\bigcup I_n = \{p(s^k(0), s^l(0), s^m(0)) \mid k, l, m \in \mathbb{N}, k + l = m\}$ as least Herbrand model for P.

Program 2

```
p(0).
p(X) :- \+ p(X).
```

The given program is level-decreasing and has B_P as its only model, which is not supported. Note that the sequence $(T^n(\emptyset))_{n\in\mathbb{N}}$ does not converge in Q.

Program 3

```
p(0).
p(s(X)) :- \+ p(X).
```

The sequence $(T^n(\emptyset))_{n\in\mathbb{N}}$ is not monotonic increasing, but converges in Q to $M = \{p(s^{2n}(0)) \mid n \in \mathbb{N}\}$, which therefore is a model for P. Moreover, M is a supported model for P by Proposition 3.10. In fact, P is strictly level-decreasing with respect to the level mapping l defined by $l(p(s^n(0))) = n$. Applying Theorem 4.16 yields M as the only supported model for P.

Program 4

```
q(0) :- \+p(X),\+p(s(X)).
p(0).
p(s(X)) :- \+p(X).
```

Define $l : B_P \to \omega + 1$ by $l(p(s^n(0))) = n$ and $l(q(s^n(0))) = \omega$ as a level mapping. By Theorem 4.12, P has a unique supported model which is the set $\{p(s^{2n}(0)) \mid n \in \mathbb{N}\}$.

Program 5

```
p(0,0).
p(s(Y),0) :- \+p(Y,X),\+p(Y,s(X)).
p(Y,s(X)) :- \+p(Y,X).
```

Define a level mapping on B_P by $l(p(s^k(0), s^j(0))) = \omega k + j$. Then P is strictly level-decreasing and hence has a unique supported model which turns out to be $\{p(0, s^{2n}(0)) \mid n \in \mathbb{N}\} \cup \{p(s^{n+1}(0), s^{2k+1}(0)) \mid k, n \in \mathbb{N}\}$.

Program 6

```
p(0) :- p(0).
p(0) :- \+ q(0).
q(0).
```

The sequence $(T^n(\emptyset))_{n\in\mathbb{N}}$ converges in Q to $M = \{p(0), q(0)\}$, which therefore is a model of P. Moreover, M is a supported model for P by Proposition 3.10. But note that M is not minimal supported, since $\{q(0)\}$ is a supported model, too. Note that P is not strictly level-decreasing but is stratified, and in this case $M_P = \{q(0)\}$.

Program 7

```
p(0).
p(X) :- p(X).
p(s(X)) :- \+p(X).
```

The sequence $(T^n(\emptyset))_{n\in\mathbb{N}}$ converges in Q to $M = B_P$, which therefore (and trivially) is a model for P. Moreover, M is a supported model for P by Proposition 3.10. But note that M is not minimal supported, since $M_1 = \{p(s^{2n}(0)) \mid n \in \mathbb{N}\}$ is a supported model, too. Note that P is not strictly level-decreasing but is semi-strictly level-decreasing, and this yields the minimal supported model M_1.

Program 8

```
r(0).
p(0) :- \+r(0).
p(s(X)) :- p(X).
q(0) :- p(X).
```

The sequence $(T^n(\emptyset))_{n\in\mathbb{N}}$ converges in Q to $M = \{r(0), q(0)\} \not\subseteq T(M) = \{r(0)\}$, which therefore is a model for P but is not supported. So it follows immediately that T can not be continuous in Q. Using the fact that P is stratified, one gets $\{r(0)\}$ as a minimal supported model for P.

Program 9

```
q(0).
q(s(s(X))) :- q(X).
p(X) :- p(X).
p(X) :- \+ q(X).
p(s(s(X))) :- \+ p(X).
```

The given program is not stratified, but is semi-strictly level-decreasing: We assign to P a level mapping l with $l(q(s^n(0))) = 0$ and $l(p(s^n(0))) = n + 1$ for every $n \in \mathbb{N}$.

Using the notation from Section 7.2, we get

$$
\begin{aligned}
I_0 &= \{q(s^{2n}(0)) \mid n \in \mathbb{N}\} \\
I_1 &= I_0 \\
I_2 &= I_1 \cup \{p(s(0))\} \\
I_3 &= I_2 \cup \{p(s^2(0))\} \\
I_4 &= I_3 \cup \{p(s^3(0))\} \\
&\vdots \\
I_n &= I_0 \cup \{p(s^m(0)) \mid 0 < m < n\}
\end{aligned}
$$

Therefore we get $I_{[P]} = \{q(s^{2n}(0)), p(s^{n+1}(0)) \mid n \in \mathbb{N}\}$ as a supported minimal model for P. The sequence $(T^n(\emptyset))_{n\in\mathbb{N}}$ converges in Q to $M = \{q(s^{2n}(0)), p(s^n(0)) \mid n \in \mathbb{N}\}$, which is a supported model for P since by Proposition 3.10, T is continuous in Q. Note that M is not a minimal model since $I_{[P]}$ is strictly smaller.

Program 10

```
p(s(0)) :- \+ q(0).
r(X):- p(X).
p(X):- r(X).
q(0).
```

Note that the sequence $(T^n(\emptyset))_{n\in\mathbb{N}}$ does not converge in the Topology Q. In fact, $T^{2n+1}(\emptyset) = \{q(0), r(s(0))\}$ and $T^{2n+2}(\emptyset) = \{q(0), p(s(0))\}$ for all $n \in \mathbb{N}$. But the sequence $(T^n(\{q(0)\}))_{n\in\mathbb{N}}$ does converge, becoming constant at $M = \{p(0)\}$ after the second step. By Proposition 3.10, M is a supported model for P. The approach using the fact that P is stratified yields the same model and states additionally that M is minimal.

Program 11

```
q(X) :- \+ r(X).
q(X) :- q(s(X)).
r(0).
r(s(X)) :- r(X).
```

The sequence $(T^n(\emptyset))_{n\in\mathbb{N}}$ converges in Q to $M = B_P$, which therefore (and trivially) is a model for P. Moreover, M is a supported model of P by Proposition 3.10. But note that M is not minimal since $M_1 = \{r(s^n(0)) \mid n \in \mathbb{N}\}$ is a supported model, too. Note, that P is not strictly level-decreasing but is semi-strictly level-decreasing, and this also yields the minimal supported model M_1.

Program 12

```
q(0).
r(0)    :- \+q(0).
r(s(X)):- r(X).
t(X)    :- r(X).
```

```
t(X)   :- t(s(X)).
```

The sequence $(T^n(\emptyset))_{n\in\mathbb{N}}$ does not converge in Q, since

$$\begin{aligned}
T(\emptyset) &= \{q(0), r(0)\} \\
T^2(\emptyset) &= \{q(0), r(s(0)), t(0)\} \\
T^3(\emptyset) &= \{q(0), r(s(s(0))), t(s(0))\} \\
T^4(\emptyset) &= \{q(0), r(s(s(s(0)))), t(0), t(s(s(0)))\} \\
T^{n+1}(\emptyset) &= \{q(0), r(s^n(0)), t(s^{(n-1)-2k}(0)) \mid k \in \mathbb{N}, 2k \leq n-1\},
\end{aligned}$$

and therefore $t(0) \in I_n$ if and only if n is even. But P is stratified, and this fact yields $\{q(0)\}$ as minimal supported model.

Program 13

```
q(0) :- \+ p(X).
p(0).
p(s(X)):- p(X).
```

The sequence $(T^n(\emptyset))_{n\in\mathbb{N}}$ converges in Q to $M = B_P$, which therefore (and trivially) is a model for P. But M is not supported, so T cannot be continuous in Q. In fact, P is strictly level-decreasing and its unique supported model is $\{p(s^n(0)) \mid n \in \mathbb{N}\}$, which is the model gained by stratifying P. Note that the interpreter for stratified programs given in [ABW88] never terminates on input $q(0)$.

Program 14

```
q(s(s(0))).
q(X):-\+q(s(X)).
```

In fact, P would be an example of a "strictly level-increasing program." The sequence $(T^n(\emptyset))_{n\in\mathbb{N}}$ does not converge in Q. A minimal supported model for P, however, is $\{q(s^{2n}(0)) \mid n \in \mathbb{N}\}$. Another minimal fixed point of T is $\{q(0), q(s(s(0))), q(s^{2n+3}(0)) \mid n \in \mathbb{N}\}$.

B Problems

We list the problems posed in the sequel by sections.

The Atomic Topology

Problem 1 Find neccessary and sufficent syntactic conditions for convergence in Q of orbits of the immediate consequence operator.

Problem 2 In the situation of Theorem 3.8, find neccessary and sufficient conditions to ensure that M is a minimal model for P.

Problem 3 Let $I \in I_P$. Consider the sequence (I_k) defined by $I_0 := I$ and $I_{k+1} := \lim T_P^n(I_k)$. When is this construction possible? Does (I_k) converge in Q? Does it become stable after finitely many steps? Is the limit a (supported) model?

Metric and P-Metric Spaces

Problem 4 To what extent can the construction of extended ultrametric spaces out of domains, as done in Section 4.1, be reversed?

Problem 5 Examine the relationships between domains and extended ultrametric spaces.

Problem 6 To what extent can Theorem 4.12 be reversed?

Problem 7 Try to find a contructive proof of Theorem 4.4 in order to find a fixed point of the function given there in the hypothesis.

Quasi-metric Spaces

Problem 8 Is it possible to weaken the assumption of the Rutten-Smyth Theorem that f is non-expansive?

Problem 9 Connect the Rutten-Smyth Theorem with Theorem 4.4by Pries-Crampe and Ribenboim. Is there a common generalization?

Problem 10 Examine the relationships between generalized metric spaces, P-metric spaces and domains.

Problem 11 Try to find a normal logic program P together with a level mapping such that T_P is not monotonic (nor Scott-continuous) but is non-expanding in order to apply the Rutten-Smyth Theorem in non-trivial cases. Is this at all possible?

Compactness of Generalized Metric Spaces

Problem 12 Investigate the importance of CN-completeness in the context of the relationships between domains and generalized metric spaces, especially, how CNs connect up with directed sets.

Problem 13 Try to characterize the topologies underlying the notions of CN-completeness. Connect this to the topologies in [BBR96], if possible.

Problem 14 Investigate the relationships between the notion of CN-completeness for generalized metric spaces and spherical completeness for P-metric spaces, perhaps by adopting a domain-theoretical point of view.

Partitioning Programs

Problem 15 In [ABW88], an interpreter for stratified logic programs (without function symbols) was given. Does there exist an interpreter for semi-strictly level-decreasing logic programs P (including function symbols) which derives exactly $I_{[P]}$? See Program 13.

Problem 16 Can the methods from Section 4 be applied to semi-strictly level-decreasing programs by using Theorem 4.4?

Problem 17 Is it possible to extend any of the approaches discussed so far to "level-increasing programs"? See Program 14.

References

[ABW88] K.R. Apt, H.A. Blair and A. Walker, *Towards a Theory of Declarative Knowledge.* In: J. Minker (Ed.), Foundations of Deductive Databases and Logic Programming. Morgan Kaufmann Publishers Inc., Los Altos, 1988, pp. 89–148.

[Bat89] A. Batarekh, *Topological Aspects of Logic Programming.* Ph. D. Thesis, Syracuse University, June 1989.

[BS89] A. Batarekh and V.S. Subrahmanian, *The Query Topology in Logic Programming.* In: Proc. 1989 Symposium on Theoretical Aspects of Computer Science. Lecture Notes in Computer Science, Vol. 349, Springer, Berlin, 1989, pp. 375–387.

[BBR96] M.M. Bonsangue, F. van Breugel and J.J.M.M. Rutten, *Generalized metric spaces: completion, topology, and powerdomains via the Yoneda embedding.* Technical Report CS-R9636, Centrum voor Wiskunde en Informatica, Amsterdam, 1996.

[Cor96] T. Cornell, *Introduction to Prolog.* Coursenotes, Seminar für Sprachwissenschaft, Universität Tübingen, Sommersemester 1996. Available at `http://www.sfs.nphil.uni-tuebingen.de/~cornell/`.

[EFT94] H.D. Ebbinghaus, J. Flum, W. Thomas, *Mathematical Logic.* Springer, Berlin, ^{2}1994.

[Fit94] M. Fitting, *Metric Methods: Three Examples and a Theorem.* J. Logic Programming <u>21</u> (3) (1994), pp. 113–127.

[ACo80] G. Gierz, K.H. Hofmann, K. Keimel, J.D. Lawson, M.W. Mislove and D.S. Scott, *A Compendium of Continuous Lattices.* Springer, Berlin, 1980.

[Hit97] P.Hitzler, *Approximation von Fixpunkten normaler logischer Programme.* Studienarbeit in Informatik, Universität Tübingen, 1997.

[Llo88] J.W. Lloyd, *Foundations of Logic Programming.* Springer, Berlin, ^{2}1988.

[Pri90] S. Priess-Crampe, *Der Banachsche Fixpunktsatz für ultrametrische Räume.* Results in Math. <u>18</u> (1990), pp. 178–186.

[PR93] S. Priess-Crampe and P. Ribenboim, *Fixed points, combs and generalized power series.* Abh. Math. Sem. Univ. Hamburg <u>63</u> (1993), pp. 227–244.

[PR97] S. Priess-Crampe and P. Ribenboim, *Ultrametric Spaces and Logic Programming.* Preprint, October 22, 1997.

[Prz88] T.C. Przymusinski, *On the Declarative Semantics of Deductive Databases and Logic Programs.* In: J. Minker (Ed.), Foundations of Deductive Databases and Logic Programming. Morgan Kaufmann Publishers Inc., Los Altos, 1988, pp. 193–216.

[Rut96] J.J.M.M. Rutten, *Elements of generalized ultrametric domain theory.* Theoretical Computer Science 170 (1996), pp. 349–381.

[Sed95] A.K. Seda, *Topology and the Semantics of Logic Programs.* Fundamenta Informaticae 24 (4) (1995), pp. 359–386.

[Sed97] A.K. Seda, *Quasi-metrics and the Semantics of Logic Programs.* Fundamenta Informaticae 29 (1) (1997), pp. 97–117.

[SH97a] A.K. Seda and P. Hitzler, *Stratified Programs Revisited: The use of topology in computational logic.* Irish Formal Methods Special Interest Group Meeting, Trinity College Dublin, September 1997.

[SH97b] A.K. Seda and P. Hitzler, *Topology and Iterates in Computational Logic.* Invited address at the 12th Summer Conference on General Topology and its Applications.: Special Session on Topology in Computer Science, North Bay, Ontario, August 1997.

[SH97c] A.K. Seda and P. Hitzler, *Topology and Iterates in Computational Logic.* submitted, 1997.

[Smy87] M.B. Smyth, *Quasi Uniformities: Reconciling Domains with Metric Spaces.* In: M. Main, A. Melton, M. Mislove and D. Schmidt (Eds.), Mathematical Foundations of Programming Language Semantics. Lecture Notes in Computer Science, Vol. 298, Springer, Berlin, 1987, pp. 236–253.

[Smy91] M.B. Smyth, *Totally bounded spaces and compact ordered spaces as domains of computation.* In: G.M. Reed, A.W. Roscoe and R.F. Wachter (Eds.), Topology and Category Theory in Computer Science. Oxford Univ. Press, 1991, pp. 207–229.

[SLG94] V. Stoltenberg-Hansen, I. Lindström and R. Griffor: *Mathematical Theory of Domains.* Cambridge Univ. Press, 1994.

[Wie96] J. Wielemaker, *SWI-Prolog 2.7, Reference Manual.* University of Amsterdam, 1996.

[Wil70] S. Willard, *General Topology.* Addison Wesley Series in Mathematics, Addison Wesley Publishing Company Inc., Massachusetts, 1970.

Ich erkläre hiermit, die vorliegende Arbeit selbständig und nur unter Zuhilfenahme der angegebenen Quellen angefertigt zu haben.