1-2001

Generalized Metrics and Topology in Logic Programming Semantics

Pascal Hitzler
pascal.hitzler@wright.edu

Generalized Metrics and Topology in Logic Programming Semantics

by

Pascal Hitzler

Dissertation submitted for the degree of Doctor of Philosophy.

Supervisor: Dr. Anthony Karel Seda

Head of Department: Prof. Gerard Murphy

Department of Mathematics
School of Mathematics, Applied Mathematics and Statistics
Faculty of Science
National University of Ireland, University College Cork

January 2001

ür Oma

Table of Contents

List of Tables

List of Figures

Abstract

Many fixed-point theorems are essentially topological in nature. Among them are the Banach contraction mapping theorem on metric spaces and the fixed-point theorem for Scott-continuous mappings on complete partial orders. The latter theorem is fundamental in denotational semantics since semantic operators in most programming language paradigms satisfy its requirements. The use of negation in logic programming and non-monotonic reasoning, however, renders some semantic operators to be non-monotonic, hence discontinuous with respect to the Scott topology, and therefore invalidates the standard approach, so that alternative methods have to be sought. In this thesis, we investigate topological methods, including generalized metric fixed-point theorems, and their applicability to the analysis of semantic operators in logic programming and non-monotonic reasoning.

In the first part of the thesis, we present weak versions of the Banach contraction mapping theorem for single-valued and multivalued mappings, and investigate relationships between the underlying spaces. In the second part, we apply the obtained results to several semantic paradigms in logic programming and non-monotonic reasoning. These investigations will also lead to a clearer understanding of some of the relationships between these semantic paradigms and of the general topological structures which underly the behaviour of the corresponding semantic operators. We will also obtain some results related to termination properties of normal logic programs, clarify some of the relationships between different semantic approaches in non-monotonic reasoning, and will establish some results concerning the conversion of logic programs into artificial neural networks.

Acknowledgements

The main funding for the research which led to this thesis was provided by Enterprise Ireland grant SC/98/621 under the project *The Use of Topology and Analysis in Computational Logic*, suggested and directed by Dr. Anthony Karel Seda. Whilst employed under this project, the research was undertaken at the Department of Mathematics, School of Mathematics, Applied Mathematics and Statistics, University College Cork, which provided all necessary facilities.

Additional travel funding was supplied from various sources: the Association of Logic Programming for attending ICLP99; the Department of Mathematics at University College Cork for attending ECAI2000; the organizers of LPNMR99, MFCSIT2000 and IWFM98 for attending their conferences; the Computational Logic Masters programme at the Department of Computer Science, Technical University of Dresden, and the Logik und Sprachtheorie group at the Department of Computer Science, University of Tübingen, for presenting my results at their colloquia. The organizers of SCAM00 and Dagstuhl Seminar 00231 invited me to attend their events and give presentations.

A number of anonymous referees have added to my knowledge. In particular, I am grateful for pointers to the work of Rounds and Zhang.

Jan Rutten was the first to point me to early work by Prieß-Crampe and Ribenboim on generalized ultrametrics; Krzysztof Apt brought my attention to [Fag91, Fag94]; Michel Schellekens directed me to [Mat92, Mat94]; Pawel Waczkiewicz filled me in on details about partial metric spaces and provided me with lots of references; Keye Martin brought my attention to [EH98].

I am deeply indebted to my supervisor, Dr. Anthony Karel Seda, for the enormous amounts of time, effort, and red ink he spent on his student. He has set up a standard which will be difficult to meet. I also thank Tony and Martine for letting the relationship grow far beyond a professional one.

I am very glad that I was able to continue working in and for enhancement programmes in Mathematics for high-school students, and this would not have been possible without the support of Prof. Finbarr Holland, UCC, and in particular of Prof. Dr. Gudrun Kalmbach H.E., University of Ulm, who has set up the programmes in which I was able to work with her for more than five years so far.

I am very grateful for my family, my extended family, for my mother, and for their support in all matters of life.

My final thanks, and all my thoughts, go to Anne.

Chapter 0

Introduction

Through the use of the fixed-point theorem for Scott-continuous functions, Theorem 1.1.3, topological considerations naturally come into view in the area of denotational semantics. Since in most programming paradigms semantic operators are Scott-continuous, hence monotonic, this theorem yields least fixed points for these operators, and these fixed points are interpreted as the denotational semantics of the programs in question. This is also the case for logic programs without negation, called *definite* logic programs.

In order to increase expressiveness and flexibility, however, it is desirable that negation may be used in logic programming. Standard semantic operators in this paradigm, though, are either not monotonic or, if they are monotonic, they are not Scott-continuous, hence do not in general achieve their least fixed points as the limit of a sequence of iterations as in the Scott-continuous case. The above mentioned approach using Theorem 1.1.3 is therefore invalid and other methods have to be sought, which include (1) the use of alternative semantic operators as e.g. in [Fit85, GRS91, GL88, HS99a], (2) restricting the syntax of the programs under consideration as e.g. in [ABW88, Cav89, Prz88, SH97], and (3) applying alternative fixed-point theorems as e.g. in [Fit85, KKM93, KM98, PCR00c, HS00]. We will touch all three approaches in this thesis while our main focus is on (3).

In the case that a semantic operator is monotonic, but not Scott-continuous, then a theorem for monotonic operators on chain-complete partial orders, Theorem 1.1.7, is the main alternative and has indeed been employed in the context of logic programming and non-monotonic reasoning, e.g. for the Fitting semantics [Fit85], cf. Chapter 6, and for the well-founded semantics [GRS91]. Some semantic operators, however, among them the immediate consequence operator and the Gelfond-Lifschitz operator [GL91], are non-monotonic and neither Theorem 1.1.3 nor Theorem 1.1.7 can be applied. A natural alternative fixed-point theorem in this case is the Banach contraction mapping theorem, Theorem 1.2.2, on metric spaces.

Since it is not a priori clear whether the spaces on which the semantic operators act are metrizable in a way such that the operators are contractions and satisfy the hypotheses of the Banach contraction mapping theorem, it is natural to ask for fixed-point theorems which are more general, i.e. act on generalized metric spaces. The development of such fixed-point theorems, the analysis of the respective underlying spaces, and investigations concerning their applicability to logic programming semantics form the heart of this thesis.

There are several ways how to generalize the notion of a metric such that a version of the Banach contraction mapping theorem can be retained, including generalized ultrametrics, quasimetrics and dislocated metrics.

Generalized ultrametrics have their origin in valuation theory, and differ from conventional ultrametrics in that the distance function maps not into the reals but into a more general partially ordered set. A number of fixed-point theorems for these spaces have been obtained and been introduced to the area of logic programming [PC90, PCR93, KKM93, SH97, BMPC99, HS99b, PCR00c, PCR00b, PCR00a], cf. also Theorem 1.3.4.

Quasimetrics [Smy91, BvBR96, Rut96], and quasi-uniformities [FL82, Smy87], which are non-symmetric distances, have recently been studied extensively in the Topology in Computer Science community. Due to their strong relationships with order structures, a fixed point theorem which reconciles Theorems 1.1.3 and 1.2.2 has been obtained [Smy87, Rut96], cf. Theorem 1.6.3. Logic programming semantics in the context of quasimetrics was studied in [Sed97, HS99c].

Dislocated metrics were studied under the notion of *metric domains* in [Mat86], where also a fixed-point theorem was given which generalizes the Banach contraction mapping theorem, cf. Theorem 1.4.6. They differ from conventional metrics in that the distance between a point and itself may be non-zero. The slightly stronger notions of partial and weak partial metrics have recently been studied further [Mat92, Mat94, O'N95, EH98, Hec99, Wac00].

Apart from the quest for generalized metric fixed-point theorems which can be applied to the semantic analysis of logic programs, some investigations using general topological approaches have been undertaken in the literature. This can be traced back to [Bat89, BS89b, BS89a], where the query topology on the space of all Herbrand interpretations was introduced. This topology was later on generalized to arbitrary preinterpretations [Sed95] and called the atomic topology. The atomic topology is a Cantor topology and can be characterized using logical notions, and it sems to be a very appropriate topology for normal logic programs and the results presented in this thesis support this claim. In fact, all models obtained by iterating non-monotonic operators in this thesis are limits in the atomic topology of these iterates.

Topological approaches to the fixed-point semantics of normal logic programs enable us to better understand the behaviour of semantic operators which arise in this context. In fact, it is clear that a (topological) space of interpretations together with such an operator can be understood as a topological dynamical system, in a naive sense. Such a point of view was hinted at in [SH97, SH99], but further results remain to be obtained, and this presents a whole bundle of new projects. We will not follow this line of thought here but refer the reader to [BDJ$^+$99] for motivational background.

Topological results in logic programming semantics also allow us to establish theoretical relationships between the theories of logic programming and of artificial neural networks [HK94, HSK99]. We present only some basic results in Chapter 9, and the study of these relationships again presents a project in its own right.

From a more general perspective, topological investigations in theoretical computer science are a natural tool to build a bridge between discrete and continuous paradigms, which is an object of study in many fields right now. The author hopes that the work presented in this thesis will be a valuable contribution to this discussion.

Some of the work in this thesis has already been presented at conferences and workshops, see e.g. [HS99a, HS99b, HS99c, HS00, SH97, SH99]. All the material has been rearranged, expanded, and brought into a more general context. All results in this thesis which are not my own are indicated as such by giving reference to the literature.

0.1 Structure of the Thesis

The thesis is divided into two parts.

Part I contains an overview of fixed-point theorems on generalized metric spaces, both for single-valued (Chapter 1) and for multivalued mappings (Chapter 2), and a discussion of relationships between underlying spaces (Chapter 3). This part assumes no knowledge in logic programming and should be of independent interest.

Part II focuses on applications of results from Part I and some other results related to logic programming semantics. After some general considerations on topological structures for normal logic programs (Chapter 4), we discuss several semantic paradigms, including the supported model semantics (Chapter 5), some semantic approaches related to the Fitting semantics (Chapter 6), the stable model semantics (Chapter 7), and the perfect and weakly perfect model semantics (Chapter 8). After some considerations concerning relationships between logic programming and artificial neural networks (Chapter 9), we close with some general conclusions (Chapter 10).

In Chapters 1 and 2, we present fixed-point theorems for single-valued and multivalued mappings on generalized metrics. Although most of these theorems are already known from the literature, we include new alternative proofs and some general investigations concerning the underlying spaces.

Chapter 3 investigates possibilities for conversion between some of the spaces from Chapters 1 and 2. We obtain new alternative proofs for some of the fixed-point theorems of the earlier chapters, a deeper insight into their relationships, and general methods for casting spaces of interpretations into generalized metrics, which will be of use in the second part of the thesis.

Chapter 4 reviews the Scott topology and the atomic topology on spaces of interpretations. The atomic topology is then generalized to many-valued logics leading to a very general framework for topological investigations of many-valued semantic operators.

In Chapter 5, we focus on the supported model semantics and in particular on uniquely determined programs, i.e. programs which have unique supported models. Step-by-step we relax syntactical and semi-syntactical conditions, leading to a hierarchy of classes of programs generalizing the acyclic programs. As these classes become more general we in turn apply more and more general fixed-point theorems from Chapter 1, each application leading to a unique fixed-point for the investigated programs, and to methods for obtaining these as topological limits.

An approach using three-valued logics in the style of [Fit85] is employed in Chapter 6. Again, we obtain a hierarchy of classes of programs which is shown to coincide with the one presented in Chapter 5.

Chapter 7 investigates the stable model semantics, both in the disjunctive and the non-disjunctive case. Relationships between the stable model semantics and the supported model semantics are obtained, and a multivalued fixed-point theorem from Chapter 2 is applied.

The perfect and the weakly perfect model semantics are studied from a topological point of view in Chapter 8. The classes described in Chapters 5 and 6 are located with respect to these semantics and generalized.

The main body of the thesis closes in Chapter 9 where relationships between logic programs and artificial neural networks, using topological methods, are studied. In particular, we address the problem of converting normal logic programs into neural networks.

Each chapter contains a *Summary and Further Work* section at the end, and final conclusions will be given in Chapter 10. We proceed now with some preliminaries and notation.

0.2 Notation

Most of the notation and notions which appear in the thesis will be introduced in the main text when they are needed for the first time. For easy reference, an index is included at the end of the thesis, which contains pointers to the definitions. We note that some of the terminology will be overloaded, i.e. the same notion may have slightly different meanings in different contexts, to keep consistency with the literature. This should pose no particular problem if care is taken as to which kind of space one is currently working with. It will be convenient now to make some general comments on notation and conventions which will be employed in the sequel.

The set of natural numbers will be denoted by $\mathbb{N}$, and of real numbers by $\mathbb{R}$; by $\mathbb{R}_0^+$ we denote the set of all positive real numbers including zero. Ordinals will usually be denoted by Greek letters, and the first infinite ordinal by ω. Each ordinal is identified with the set of all its predecessors, i.e. for each ordinal α we have $\{\beta \mid \beta < \alpha\} = \{\beta \mid \beta \in \alpha\}$, and using this convention, we identify ω with $\mathbb{N}$. If α is a successor ordinal, we denote its predecessor by $\alpha - 1$, and the successor of an arbitrary ordinal α will be denoted by $\alpha + 1$.

If $f : X \to Y$ is a function and $A \subseteq X$, we set $f(A) = \{f(a) \mid a \in A\}$.

Ordinal powers of functions are defined as follows. Let $f : X \to X$ be a function on a set X, and let $x \in X$. We define $f~(x) = x$ and for each successor ordinal $\alpha+1$ we define $f^{\alpha+1}(x) = f(f^{\alpha}(x))$. If α is a limit ordinal, we will require several methods in the sequel how to define $f^{\alpha}(x)$, and we will define these on the spot for the respective context. Thus, if we define $f^{\alpha}(x)$ for each limit ordinal α, it will be unambiguous in each case what all ordinal powers of the given function f are.

A partially ordered set $(\Lambda, \leq)$ is *directed* if for all $x, y \in \Lambda$ there exists $z \in \Lambda$ such that $x \leq z$ and $y \leq z$. For each $\beta \in \Lambda$ we define $\uparrow\beta = \{\lambda \in \Lambda \mid \beta \leq \lambda\}$.

A *net* $(x_\lambda)_{\lambda \in \Lambda}$ is a net in the topological sense i.e. the index set Λ is directed, and the index set will be omitted, i.e. the net will be written as (x_λ) or even just x_λ when the meaning is clear from the context; the notation $(x_\lambda)_\Lambda$ will also be used. For each $\lambda \in \Lambda$, x_λ is called an *element* of the net $(x_\lambda)_\Lambda$. Given a net $(x_\lambda)_\Lambda$ and an element $\beta \in \Lambda$, we call the subnet $(x_\lambda)_{\lambda \geq \beta} = (x_\lambda)_{\uparrow\beta}$ of $(x_\lambda)_\Lambda$ a *tail* of $(x_\lambda)_\Lambda$.

A net with index set equal to ω, or equivalently $\mathbb{N}$, is called a *sequence*. A *transfinite sequence* is a net where the index set is an ordinal. A *chain* is a linearly ordered family of elements of a given partially ordered set. An *ω-chain* is a sequence which is a chain.

If X is a set and $f : X \to X$ is a function then each $x \in X$ with $f(x) = x$ is called a *fixed point* of f. If X carries a partial order $\leq$, then each $x \in X$ with $f(x) \leq x$ is called a *pre-fixed point* of f. If f is a mapping from X to the powerset 2^X of X, then f is called a *multivalued mapping* on X. In this case, each $x \in X$ with $x \in f(x)$ is called a *fixed point* of f. Each single-valued mapping f on a set X can be identified with a multivalued mapping by identifying each $f(x) \in X$ with $\{f(x)\} \in 2^X$. We will assume throughout that multivalued mappings are *non-empty*, i.e. that $f(x) \neq \emptyset$ for all $x \in X$.

A *distance function* on a set X is a mapping from $X \times X$ to a given set Λ, where Λ will always be either the set of real numbers $\mathbb{R}$ or some partially ordered set. A *generalized ultrametric* is a distance function which maps into a partially ordered set and satisfies some specific further conditions which will be given in Definition 1.3.1. In contrast to this, a *generalized metric* is a distance function which either maps into $\mathbb{R}$ and satisfies the triangle inequality (Miv) of Definition 1.2.1, or which maps into a partially ordered set and satisfies the corresponding strong triangle inequality (Uiv) of Definition 1.3.1. This usage of the term *generalized* is not entirely consistent, but is adopted here in order to compromise between established notation and convenience: The term *generalized ultrametric* refers to a specific structure (Definition 1.3.1) and is standard. The term *generalized metric* refers to all notions appearing in this thesis which can be understood as generalizations of metrics (or ultrametrics) in a naive sense. This contrasts to the use of this notion in some of the literature where the term *generalized metric* refers to quasi-pseudo-metrics only, see Definition 1.2.1.

We will usually denote distance functions with d, unless the requirement that self-distances of points are zero is dropped ((Mi) in Definition 1.2.1, (Uii) in Definition 1.3.1), in which case we will usually denote them by ϱ to help the reader. All generalized metric spaces are supposed to be non-empty.

Some of the major fixed-point theorems will be given names for convenience. Theorem 1.1.3, for example, will be called the Kleene theorem, and it will be referred to as either the *Kleene theorem*, or the *Kleene theorem, Theorem 1.1.3*, or more simply, with a slight abuse of language, the *Kleene theorem 1.1.3*. Other named theorems will be referred to analogously. It is not claimed that the names given to theorems in this thesis are historically correct, see [LNS82].

Notation for logic programming basically follows [Llo88].

Given a first order language $\mathcal{L}$, a *normal logic program*, referred to as *logic program* or simply *program*, is a finite set of *clauses* of the form

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

where $n \in \mathbb{N}$ may differ between clauses, A is an atom in $\mathcal{L}$ and $L_1, \ldots, L_n$ are literals, i.e. atoms or negated atoms, in $\mathcal{L}$. As is customary in logic programming, we will write such a clause as

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

and A is called the *head* of the clause, each L_i is called a *body literal* of the clause and their conjunction $L_1, \ldots, L_n$ is called the *body* of the clause. We allow $n = 0$, by an abuse of notation, in which case the body is empty and the clause is called a *unit clause* or a *fact*. We will occasionally use the notation $A \leftarrow$ `body` for clauses, i.e. `body` in this case stands for the conjunction of the body literals of the clause. If no negation symbol occurs in a logic program, it is called a *definite* or *positive logic program*. A variable in a clause is said to be *local* if it occurs in the body of the clause, but not in the corresponding head.

0.2.1 Program The following is an example of a normal logic program:

$$\begin{aligned}
\mathrm{distlist}([\,]) &\leftarrow \\
\mathrm{distlist}([H|T]) &\leftarrow \mathrm{distlist}(T), \neg\, \mathrm{member}(H,T) \\
\mathrm{member}(X,[X|T]) &\leftarrow \\
\mathrm{member}(X,[H|T]) &\leftarrow \mathrm{member}(X,T)
\end{aligned}$$

In the above example, uppercase letters denote variable symbols. The constant symbol $[\,]$ is interpreted as the empty list and $[H|T]$ as a list with head H and tail T, hence $[.|.]$ is a function symbol with arity 2. The intended meaning of the program is that $\mathrm{member}(x,l)$ is true if x is an element of the list l, and $\mathrm{distlist}(l)$

is true if l is a list of mutually distinct elements. Under a logic programming system like Prolog, the above program can indeed be used to check whether a list consists of mutually distinct elements.

Given a preinterpretation J for a first order language $\mathcal{L}$ underlying a given logic program P, the set of all ground instances of atoms occurring in P, under J, will be denoted by $B_{P,J}$, or just by B_P if this will cause no misunderstandings. In the case of J being the Herbrand preinterpretation corresponding to $\mathcal{L}$, we will call B_P the *Herbrand base* of P. The set of all ground instances of clauses in P (with respect to an arbitrary, but fixed preinterpretation J) will be denoted by ground(P). The set of all interpretations of P under J will be denoted by $I_{P,J}$ or simply by I_P. Each $I \in I_P$ is identified with the set of all ground atoms which are true with respect to I, i.e. we identify I_P with the power set 2^{B_P}, and for each $I \in I_P$ we have $\{A \in B_P \mid I \models A\} = \{A \in B_P \mid A \in I\}$. Due to this identification, the set I_P carries a natural order structure, namely set-inclusion. If I is an interpretation of a program P, we denote its *complement* $B_P \setminus I$ by ${}^cI$.

Given a program P, the *language underlying* P is the first order language with constant, function, and predicate symbols being, respectively, the constant, function, and predicate symbols occurring in P; if no constant symbol is present, however, we add the symbol 0 as a constant symbol to the language. If we state that J is an (arbitrary) preinterpretation it is always assumed that J is suitable for the program in question, i.e. it is a preinterpretation for the language underlying the program.

0.2.2 Definition Given a logic progam P and a preinterpretation J, we define the *single-step operator* or *immediate consequence operator* $T_{P,J}$, or simply T_P, as a mapping from I_P to I_P as follows. For each $I \in I_P$ we set $T_P(I)$ to be the set of all $A \in B_P$ for which there exists a clause $A \leftarrow L_1, \dots, L_n$ in ground(P), such that $I \models L_1$ L_n.

The usefulness of the operator T_P in the semantic analysis of logic programs rests on the fact that the models of P are exactly the pre-fixed points of T_P [Llo88]. A model of P is called a *supported model* (or model of the *Clark completion*[1] of P [Cla78]) if it is a fixed point of T_P [ABW88].

A *level mapping* for a program P is a mapping $l : B_P \to \alpha$, where α is an ordinal. If $\alpha = \omega$, l is called an *ω-level mapping*. We always assume that a level mapping is extended to ground literals by setting $l(\neg A) = l(A)$ for all $A \in B_P$.

We finally remark that the term *semantics* in this thesis refers to *declarative* or *denotational* semantics, and we will use the term *procedural semantics* if we want to refer to the procedural, or operational aspects.

[1] The correspondence between supported models and models of the Clark completion is in fact via a standard identification.

Part I

Fixed-point Theorems on Generalized Metric Spaces

Chapter 1

Fixed-point Theorems for Single-valued Mappings

We present fixed-point theorems which will be applied in Part II of the thesis, and some further results. Section 1.1 contains the fundamental fixed-point theorems on partially ordered sets which play a central role in the denotational semantics of logic programs. Section 1.2 introduces generalized metrics where the distance functions map into the real numbers, and recalls the Banach contraction mapping theorem. Section 1.3 recalls the Prieß-Crampe and Ribenboim theorem on generalized ultrametric spaces, including an alternative proof, and discusses its relation to the Banach contraction mapping theorem. Section 1.4 discusses the corresponding fixed-point theorem by Matthews on dislocated metrics and some topological matters concerning these spaces. The latter two theorems are then merged in Section 1.5, and finally, in Section 1.6, the Rutten-Smyth theorem on quasimetrics is discussed.

1.1 Partial Orders

The set of all interpretations of a logic program, with respect to a given preinterpretation, is essentially a powerset. With the subset ordering, it becomes a complete lattice. We present two classical fixed-point theorems on weaker order structures, which play a fundamental role in logic programming semantics.

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

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

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

1.1.2 Definition Let D and E be ω-cpos and let $f : D \to E$ be 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 it is monotonic and for every ω-chain $a \leq a_1 \leq \dots$ we have $f(\sup_{i\in\mathbb{N}} a_i) = \sup_{i\in\mathbb{N}} f(a_i)$.

The following theorem is of fundamental importance in the theory of denotational semantics.

1.1.3 Theorem (Kleene theorem) 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, $a = \sup_{n\in\mathbb{N}} f^n(\bot)$.

Proof: We sketch the well-known proof. The sequence $(f^n(\bot))_{n\in\mathbb{N}}$ is an increasing chain, hence has a supremum a. By continuity of f, we obtain $f(a) = a$, hence a is a fixed point which turns out to be least since for any other fixed point b of f we obtain $f^n(\bot) \leq b$ by an easy induction argument. ■

If P is a definite logic program, then the hypotheses of Theorem 1.1.3 are satisfied by the operator T_P, which is well-known [Llo88]. In Part II of the thesis, we will study programs with negation, in which case semantic operators are not necessarily ω-continuous, and sometimes not even monotonic, so that Theorem 1.1.3 cannot be applied.

The notion of ω-continuity is a weak version of Scott-continuity, which is usually defined on Scott-Ershov domains, introduced next.

1.1.4 Definition A partially ordered set $(D, \sqsubseteq)$ is called a *(Scott-Ershov) domain* with set D_c of *compact elements* (see [SHLG94]), if the following conditions hold:

(i) $(D, \sqsubseteq)$ is a *complete partial order* (*cpo*), that is, D has a bottom element $\bot$, and the supremum $\sup A$ exists for all directed subsets A of D.

(ii) The elements $a \in D_c$ are characterized as follows: whenever A is directed and $a \sqsubseteq \sup A$, then $a \sqsubseteq x$ for some $x \in A$.

(iii) For each $x \in D$, the set $\operatorname{approx}(x) = \{a \in D_c \mid a \sqsubseteq x\}$ is directed and $x = \sup \operatorname{approx}(x)$ (this property is called *algebraicity* of D).

(iv) If the subset A of D is consistent (there exists $x \in D$ such that $a \sqsubseteq x$ for all $a \in A$), then $\sup A$ exists in D (this property is called *consistent completeness* of D).

We will usually denote the order relation by $\sqsubseteq$ if the order structure under consideration is a domain.

Several important facts emerge from these conditions, including the existence of function spaces (the category of domains is cartesian closed). Moreover, the compact elements provide an abstract notion of computability. Domains were introduced independently by D.S. Scott and Y.L. Ershov as a means of providing structures for modelling computation, and to provide spaces to support the denotational semantics approach to understanding programming languages, see [SHLG94].

The standard topology on a domain is the *Scott topology*, defined as follows.

1.1.5 Definition Let $(D, \sqsubseteq)$ be a domain. The set $\{\uparrow c \mid c \in D_{\mathrm{c}}\}$ is a base for a topology, called the *Scott topology* on D. A function $f : D \to D$ is called *Scott-continuous* if it is continuous with respect to the Scott topology. Equivalently (see [SHLG94]), f is Scott continuous if and only if it is monotonic and for each directed set $A \quad D$ we have $\sup f(A) = f(\sup A)$.

It is clear that every domain is a cpo and every cpo is an ω-cpo. Likewise, every Scott-continuous function on a domain is also ω-continuous. Theorem 1.1.3 is often stated in less general form on domains for Scott-continuous functions, or even on complete lattices.

If an operator is monotonic but not Scott-continuous, the existence of a least fixed point can still be guaranteed, although not as the limit of an ω-chain.

1.1.6 Definition A partial order D is called *chain-complete* if every chain in D has a supremum.

1.1.7 Theorem (Knaster-Tarski theorem) Let $(D, \leq)$ be a chain-complete partial order, let $f : D \to D$ be monotonic, and let $a \in D$ be such that $a \leq f(a)$. Then f has a least fixed point x above a and there exists a least ordinal γ such that $f^{\gamma}(a) = x$.

Proof: We sketch the well-known proof. For any limit ordinal α define $f^{\alpha}(a) = \sup\{f^{\beta}(a) \mid \beta \quad \alpha\}$, from which we obtain a transfinite increasing sequence of iterates of f. Let γ be an ordinal whose cardinality is greater than the cardinality of D. Then $f^{\gamma}(a)$ must be a fixed point of f which is above a. ∎

We find it convenient to introduce names for Theorems 1.1.3 and 1.1.7, although this is not always done. We will call Theorem 1.1.3 the *Kleene theorem*, and Theorem 1.1.7 the *Knaster-Tarski theorem*. We would like to note that this notation is not standard, but will be very convenient in the sequel.

1.2 Metrics

We introduce some notions of generalized metrics and state the Banach contraction mapping theorem for conventional metrics.

1.2.1 Definition Let X be a set and let $\varrho : X \times X \to \mathbb{R}^{+}$ be a function, called a *distance function*. Consider the following conditions:

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

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

notion satisfies	(Mi)	(Mii)	(Miii)	(Miv)	(Miv')
metric	x	x	x	x	
ultrametric	x	x	x	(x)	x
pseudometric	x		x	x	
pseudo-ultrametric	x		x	(x)	x
quasimetric	x	x		x	
quasi-ultrametric	x	x		(x)	x
dislocated metric		x	x	x	
dislocated ultrametric		x	x	(x)	x
dislocated quasimetric		x		x	
dislocated quasi-ultrametric		x		(x)	x
quasi-pseudo-metric	x			x	
quasi-pseudo-ultrametric	x			(x)	x

Table 1.1: Generalized metrics: Definition 1.2.1.

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

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

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

If ϱ satisfies conditions (Mi) to (Miv), then it is called a *metric*. If it satisfies conditions (Mi), (Miii) and (Miv), it is called a *pseudometric*. If it satisfies (Mii), (Miii) and (Miv), we will call it a *dislocated metric* (or simply *d-metric*). A *quasi-metric* satisfies conditions (Mi), (Mii) and (Miv). Condition (Miv) will be called the *triangle inequality*. If a (pseudo-, quasi-, d-) metric satisfies the *strong triangle inequality* (Miv$'$), then it is called a (pseudo-, quasi-, d-) *ultrametric*. These definitions are listed in Table 1.1; an x indicates that the respective condition is satisfied. (x) indicates that the respective condition is automatically satisfied.

1.2.2 Theorem (Banach contraction mapping theorem) Let (X, d) be a complete metric space, $0 \leq \lambda \quad 1$ and let $f : X \to X$ be a function which is a *contraction* with *contractivity factor* λ, i.e. satisfies $d(f(x), f(y)) \leq \lambda d(x, y)$ for all $x, y \in X$ (with $x = y$). Then f has a unique fixed point which can be obtained as the limit of the sequence $(f^n(x))$ for any $x \in X$.

Proof: We sketch the well-known proof. For any $x \in X$, the sequence $(f^n(x))$ is a Cauchy sequence which converges to a unique limit x by completeness of the space. Since f is a contraction, it is continuous, hence x is a fixed point of f, and is easily shown to be unique. ■

It is well-known that the requirement $\lambda < 1$ cannot be relaxed in general, as can be seen from the function

$$f : \mathbb{R} \to \mathbb{R} : x \mapsto \begin{cases} x \quad \frac{1}{x} & \text{for } x \geq 1 \\ 2 & \text{otherwise,} \end{cases}$$

which satisfies the condition $d(f(x), f(y)) \quad d(x, y)$ for all $x, y \in \mathbb{R}$ with $x = y$, where d is the natural metric on $\mathbb{R}$, but has no fixed point since $f(x) > x$ for all $x \in \mathbb{R}$. If X is compact, however, the requirement on λ can be relaxed.

1.2.3 Theorem Let (X, d) be a compact metric space and let $f : X \to X$ be a function which is *strictly contracting*, i.e. satisfies $d(f(x), f(y)) \quad d(x, y)$ for all $x, y \in X$ with $x = y$. Then f has a unique fixed point.

Proof: The function $\overline{d}(x) = d(x, f(x))$ is continuous since f is continuous. Therefore, it achieves a minimum m on X. Assume $\overline{d}(x\ ) = m > 0$. Then $\overline{d}(f(x\ )) = d(f(x\ ), f(f(x\ ))) \quad d(x\ , f(x\ )) = \overline{d}(x\ ) = m$ which is a contradiction. Hence $m = 0$ and f has a fixed point.

Assume x and y are fixed points of f and $x = y$. Then $d(x, y) = d(f(x), f(y)) \quad d(x, y)$ which is a contradiction. Therefore, the fixed point of f is unique. ■

The above result can be found e.g. in [DG82].

1.3 Generalized Ultrametrics

The origin of generalized ultrametrics lies in valuation theory. They differ from conventional metrics in that the distance function takes values in general partially ordered sets instead of the real numbers. We introduce generalized ultrametrics and dislocated generalized ultrametrics, state the Prieß-Crampe and Ribenboim theorem 1.3.4 which is the analogue on these spaces of the Banach contraction mapping theorem 1.2.2, and study the notion of spherical completeness of generalized ultrametric spaces in how it relates to completeness and compactness for conventional metrics. We also give a constructive proof of a part of the Prieß-Crampe and Ribenboim theorem.

1.3.1 Definition Let X be a set and let Γ be a partially ordered set with least element 0. We call (X, ϱ, Γ) (or simply (X, ϱ)) a *generalized ultrametric space* (*gum*) if $\varrho : X \times X \to \Gamma$ is a function such that for all $x, y, z \in X$ and all $\gamma \in \Gamma$ we have:

(Ui) $\varrho(x, y) = 0$ implies $x = y$.

(Uii) $\varrho(x, x) = 0$.

(Uiii) $\varrho(x, y) = \varrho(y, x)$.

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

If ϱ satisfies conditions (Ui), (Uiii) and (Uiv), but not necessarily (Uii), we call (X, ϱ) a *dislocated generalized ultrametric space* or simply a *d-gum space*, cf. Table 1.2. Condition (Uiv) will be called the *strong triangle inequality for gums*.

We will occasionally refer to the set Γ as the *distance set* of (X, ϱ).

notion satisfies	(Ui)	(Uii)	(Uiii)	(Uiv)
generalized ultrametric (gum)	x	x	x	x
dislocated generalized ultrametric (d-gum)	x		x	x

Table 1.2: (Dislocated) generalized ultrametrics: Definition 1.3.1.

It is clear that every (conventional) ultrametric space is also a generalized ultrametric space.

The following definitions prepare Theorem 1.3.4 and are taken from [PCR00a].

1.3.2 Definition Let (X, ϱ, Γ) be a d-gum space. For $0 = \gamma \in \Gamma$ and $x \in X$, the set $B_\gamma(x) = \{y \in X \mid \varrho(x,y) \leq \gamma\}$ is called a (γ-)*ball* in X with *centre* or *midpoint* x. A d-gum space is called *spherically complete* if, for any chain $\mathcal{C}$, with respect to set-inclusion, of non-empty balls in X, we have $\bigcap \mathcal{C} = \emptyset$. A function $f : X \to X$ is called

(1) *non-expanding* if $\varrho(f(x), f(y)) \leq \varrho(x,y)$ for all $x, y \in X$,

(2) *strictly contracting on orbits* if $\varrho(f^2(x), f(x)) \quad \varrho(f(x), x)$ for every $x \in X$ with $x = f(x)$, and

(3) *strictly contracting* if $\varrho(f(x), f(y)) \quad (x, y)$ for all $x, y \in X$ with $x = y$.

The requirement in the definition of spherical completeness that all balls are non-empty can be dropped when working in a gum instead of a d-gum, since in the first case all balls are always non-empty.

We will need the following observations, which are well-known for ordinary ultrametric spaces, see [PCR93].

1.3.3 Lemma Let (X, ϱ, Γ) be a d-gum space. For $\alpha, \beta \in \Gamma$ and $x, y \in X$ the following statements hold.

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

(2) If $B_\alpha(x) \cap B_\alpha(y) = \emptyset$, then $B_\alpha(x) = B_\alpha(y)$. In particular, each element of a ball is also its centre.

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

Proof: Let $a \in B_\alpha(x)$ and $b \in B_\alpha(x) \cap B_\beta(y)$. Then $\varrho(a, x) \leq \alpha$ and $\varrho(b, x) \leq \alpha$, hence $\varrho(a, b) \leq \alpha \leq \beta$. Since $\varrho(b, y) \leq \beta$, we have $\varrho(a, y) \leq \beta$, hence $a \in B_\beta(y)$, which proves the first statement. The second follows by symmetry and the third by replacing $\varrho(x, y)$ by α and applying (2). ■

For the following, see [PCR00c]. We will give several alternative proofs later.

1.3.4 Theorem (Prieß-Crampe and Ribenboim theorem) Let (X, d) be a spherically complete generalized 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.

Note that every compact ultrametric space is spherically complete by the finite intersection property. The converse is not true: let X be an infinite set and take $d(x, y) = 1$ if $x = y$ and $d(x, x) = 0$ for all x. Then (X, d) is not compact but spherically complete. The relationship between spherical completeness and completeness is given by the next proposition. Similar investigations have been undertaken in [PC90] in the case of totally ordered distance sets.

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

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

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

Note also that with the notation from the second part of the proof, the successor function $n \to n \quad 1$ is strictly contracting, but does not have a fixed point. By Proposition 1.3.5 and the remarks preceding it, we obtain that the notion of spherical-completeness is strictly less general than completeness, and is strictly more general than compactness.

We will now follow a line of thought from [PC90], only slightly changed (the original version was for linearly ordered distance set), and with the proofs adapted to the more general setting.

1.3.6 Definition Let $(x_\delta)_{\delta<\varrho}$ be a (possibly transfinite) sequence of elements of a gum (X, d). Then (x_δ) is said to be *pseudo-convergent* if for all $\alpha \quad \beta \quad \gamma \quad \varrho$ we have $d(x_\beta, x_\gamma) \quad d(x_\alpha, x_\beta)$. The transfinite sequence $(\pi_\delta)_{\delta+1<\varrho}$ with $\pi_\delta = d(x_\delta, x_{\delta+1})$ is then strictly monotonic decreasing. If ϱ is a limit ordinal, then any $x \in X$ with $d(x, x_\delta) \leq \pi_\delta$ for all $\delta \quad \varrho$ is called a *pseudo-limit* of the transfinite sequence $(x_\delta)_{\delta<\varrho}$.

The space (X, d) is called *trans-complete* if every pseudo-convergent transfinite sequence $(x_\delta)_{\delta<\varrho}$, where ϱ is a limit ordinal, has a pseudo-limit in X.

1.3.7 Proposition If x is a pseudo-limit of $(x_\delta)_{\delta<\varrho}$, where ϱ is a limit ordinal, then the set of all pseudo-limits of (x_δ) is given by $\mathrm{Lim}(x_\delta) = \{z \in X \mid d(x, z) \quad \pi_\delta$ for all $\delta < \quad \varrho\}$.

Proof: Let $z \in \mathrm{Lim}(x_\delta)$. Since $d(z, x) \quad \pi_\delta$ and $d(x, x_\delta) \leq \pi_\delta$ we obtain $d(z, x_\delta) \leq \pi_\delta$ for all δ. Conversely, let z be a pseudo-limit of (x_δ). Since $d(x, x_{\delta+1}), d(z, x_{\delta+1}) \leq \pi_{\delta+1}$ for all $\delta \quad \varrho$, we obtain $d(x, z) \leq \pi_{\delta+1} \quad \pi_\delta$ for all $\delta < \quad \varrho$. ■

1.3.8 Proposition A generalized ultrametric space is spherically complete if and only if it is trans-complete.

Proof: Let X be trans-complete and let $\mathcal{B}$ be a decreasing chain of balls in X. Without loss of generality assume that $\mathcal{B}$ does not have a minimal element and is in fact strictly decreasing. Then we can select a coinitial subchain $(B_\delta)_{\delta<\varrho}$ of $\mathcal{B}$, where ϱ is a limit ordinal, i.e. $(B_\delta)_{\delta<\varrho}$ is a transfinite sequence of balls. Since this transfinite sequence is strictly decreasing, we know that for every δ there exists $x_\delta \in B_\delta \setminus B_{\delta+1}$, and the transfinite sequence $(x_\delta)_{\delta<\varrho}$ is pseudo-convergent, hence has a pseudo-limit x. Since $d(x, x_\delta) \leq d(x_\delta, x_{\delta+1})$ and $x_\delta, x_{\delta+1} \in B_\delta$ we obtain $x \in B_\delta$ for all δ, hence $x \in \quad \mathcal{B}$.

Conversely, let X be spherically complete and let (x_δ) be pseudo-convergent. Let $\pi_\delta = d(x_\delta, x_{\delta+1})$ and $B_\delta = B_{\pi_\delta}(x_\delta)$. For $\alpha \quad \beta$ we have $x_\beta \in B_\alpha \cap B_\beta$ and therefore that (B_δ) is a decreasing chain of balls by Lemma 1.3.3. By spherical completeness, there is some $x \in \quad B_\delta$ which is a pseudo-limit of (x_δ). ■

We can now give a constructive proof of the second part of Theorem 1.3.4 under the restriction that Γ is linearly ordered. The proof is inspired by [KKM93], cf. also Section 2.2.

1.3.9 Theorem Let (X, d, Γ) be a spherically complete generalized ultrametric space where Γ is linearly ordered and let $f : X \to X$ be strictly contracting on X. Then f has a unique fixed point.

Proof: Choose some $x \quad \in X$ and let $x_1 = f(x \quad)$. We inductively define a transfinite sequence as follows. Our induction hypothesis is that for all ordinals β the sequence $(x_\beta)_{\beta<\alpha}$ is pseudo-convergent. We also assume, without loss of generality, that none of the x_β is a fixed point of f.

If $\alpha = \beta \quad 1 \quad 1$ is the successor of a successor ordinal, then let $x_\alpha = f(x_{\beta+1})$. Since f is strictly contracting, the obtained sequence $(x_\beta)_{\beta\leq\alpha}$ is pseudo-convergent.

If α is a limit ordinal, then $(x_\beta)_{\beta<\alpha}$ is pseudo-convergent by the induction hypothesis. Then choose x_α to be one of its pseudo-limits, which is possible by

Proposition 1.3.8, and let $\gamma_1 \quad {}_2$. Then by the induction hypothesis <

$$d(x_{\gamma_2}, x_\alpha) \leq d(x_{\gamma_2}, x_{\gamma_2+1})$$
$$(x_{\gamma_1}, x_{\gamma_2}).$$

So the resulting sequence is also pseudo-convergent.

If $\alpha = \beta$ is the successor of a limit ordinal, where x_β is constructed as +1 in the previous paragraph, then let $x_\alpha = f(x_\beta)$. We have to show that for all $\gamma_1 \quad {}_2 \leq \beta$ we have $d(x_{\gamma_2}, x_\alpha) \quad (x_{\gamma_1}, x_{\gamma_2})$. $<\gamma$

First assume that γ_2 is a limit ordinal. For every $\gamma \quad 1 \quad \gamma_2$ we obtain $d(x_{\gamma+1}, x_\alpha) \quad d(x_\gamma, x_\beta) \leq d(x_\gamma, x_{\gamma_2})$ since f is strictly contracting and by the induction hypothesis, and $d(x_{\gamma+1}, x_{\gamma_2}) \quad d(x_\gamma, x_{\gamma_2})$ by the following argument: $d(x_{\gamma+1}, x_{\gamma_2}) \leq \pi_{\gamma+1} \quad \pi_\gamma = d(x_\gamma, x_{\gamma+1})$, hence $x_\gamma \in B_{\pi_{\gamma+1}}(x_{\gamma+1}) = B_{\pi_{\gamma+1}}(x_{\gamma_2})$ which suffices. By (Uiv) we conclude that $d(x_{\gamma_2}, x_\alpha) \quad (x_\gamma, x_{\gamma_2})$ as required.

It remains to show the case where γ_2 is a successor ordinal. We obtain

$$d(x_{\gamma_2}, x_\alpha) \quad (x_{\gamma_2-1}, x_\beta)$$
$$(x_{\gamma_1}, x_{\gamma_2})$$

since f is strictly contracting and by the induction hypothesis.

We constructed a transfinite sequence (x_α) which is pseudo-convergent. We also obtain a corresponding sequence π_α in Γ, where $\pi_\alpha = d(x_\alpha, x_{\alpha+1})$, which is strictly decreasing. If we assume that no point in (x_α) is a fixed point, then there must be an ordinal γ such that $\pi_\alpha = 0$ for all $\alpha > \gamma$, where 0 is the least element of Γ. This, however, contradicts the assumption that no point in (x_α) is a fixed point.

In order to finish the proof, we need to show uniqueness of the fixed point. Suppose y is another fixed point of f. Then $d(x, y) = d(f(x), f(y)) \quad d(x, y)$ which is a contradiction. Hence the fixed point is unique. ■

An alternative constructive proof is given in Section 1.5.

1.4 Dislocated Metrics

Dislocated metrics were studied under the name of *metric domains* in [Mat86]. We proceed now with the definitions needed for stating the Matthews theorem, which is the generalized Banach contraction mapping theorem on these spaces, that is, we will define convergence, Cauchy sequences and completeness for dislocated metrics as in [Mat86]. As it turns out, these notions can be carried over directly from conventional metrics. Then, we will investigate the topological structure underlying the notion of dislocated metric, which will lead to a proof of the Matthews theorem which is in the spirit of the proof of the Banach contraction mapping theorem.

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

1.4.2 Proposition Limits in d-metric spaces are unique.

Proof: Let x and y be limits of the sequence (x_n). By properties (Miii) and (Miv) of Definition 1.2.1, it follows that $\varrho(x, y) \leq \varrho(x_n, x) + \varrho(x_n, y) \to 0$ as $n \to$. Hence $\varrho(x, y) = 0$ and by property (Mii) of Definition 1.2.1 it follows that $x = y$. ■

1.4.3 Definition A sequence (x_n) in a d-metric space is called a *Cauchy sequence* if for each $\varepsilon > 0$ there exists $n \in \mathbb{N}$ such that for all $m, n \geq n$ we have $\varrho(x_m, x_n)$. $< \varepsilon$

1.4.4 Proposition Every convergent sequence in a d-metric space is a Cauchy sequence.

Proof: Let (x_n) be a sequence which converges to some x, and let $\varepsilon > 0$ be arbitrarily chosen. Then there exists $n \in \mathbb{N}$ with $\varrho(x_n, x)$ $\frac{\varepsilon}{2}$ for all $n \geq n$. For $m, n \geq n$ we then obtain $\varrho(x_m, x_n) \leq \varrho(x_m, x) + \varrho(x, x_n)$ 2 $\frac{\varepsilon}{2} = \varepsilon$. Hence (x_n) is a Cauchy sequence. ■

1.4.5 Definition A d-metric space (X, ϱ) is called *complete* if every Cauchy sequence in X converges with respect to ϱ. A function $f : X \to X$ is called a *contraction* if there exists $0 \leq \lambda$ 1 such that $\varrho(f(x), f(y)) \leq \lambda \varrho(x, y)$ for all $x, y \in X$.

1.4.6 Theorem (Matthews theorem) Let (X, ϱ) be a complete d-metric space and let $f : X \to X$ be a contraction. Then f has a unique fixed point.

A proof of this theorem was given in [Mat86], and we will from now on refer to it as the *Matthews theorem*. We will give an alternative proof later which is more in the spirit of the proof of the original Banach contraction mapping theorem.

We will now investigate a topological point of view of dislocated metrics following the outline given by the definitions at the beginning of this section. Since constant sequences do not in general converge in d-metric spaces, a conventional topological approach is not feasible, and notions of neighbourhoods, convergence and continuity will have to be modified.

Dislocated Neighbourhoods

1.4.7 Definition An (*open* ε-)*ball* in a d-metric space (X, ϱ) with *centre* $x \in X$ is a set $B_\varepsilon(x) = \{y \in X \mid \varrho(x, y) \quad \}$ where $\varepsilon > 0$.

Note that balls may be empty in d-metric spaces. In fact, the above definition of ball does not imply that the centre of a ball is contained in the ball itself: the point may be *dislocated* from the ball, and hence our usage of the term "dislocated".

1.4.8 Proposition Let (X, ϱ) be a d-metric space.

(a) The following three conditions are equivalent:

(i) For all $x \in X$, we have $\varrho(x, x) = 0$.

(ii) ϱ is a metric.

(iii) For all $x \in X$ and all $\varepsilon > 0$, we have $B_\varepsilon(x) = \emptyset$.

(b) The space (X', ϱ), where $X' = \{x \in X \mid \varrho(x, x) = 0\}$, is a metric space.

Proof: (a) That (i) implies (ii) is obvious, as is (ii) implies (iii). We show (iii) implies (i). Since $B_\varepsilon(x) = \emptyset$ for all $\varepsilon > 0$, there exists, for each $\varepsilon > 0$, some $y \in X$ with $\varrho(x, y) \quad \varepsilon$. But, for all $y \in X$, we have $\varrho(x, x) \leq 2 \quad \varrho(x, y)$, and hence $\varrho(x, x) \quad$ for all $\varepsilon > 0$. Therefore, $\varrho(x, x) = 0$. $<\varepsilon$
(b) Obviously, (X', ϱ) is a d-metric space. The assertion now follows immediately from (a). ■

We proceed with the investigation of dislocated metrics from a topological point of view.

1.4.9 Definition Let X be a set. A relation $\circ \quad X \times \mathcal{P}(X)$ (written infix) is called a *d-membership relation* (*on* X) if it satisfies the following property for all $x \in X$ and $A, B \quad X$:

$$x \lessdot A \text{ and } A \quad B \text{ implies } x \lessdot B. \tag{1.1}$$

We say x is below A” if $x \lessdot A$.

The below”-relation is a generalization of the membership relation from set-theory, which will allow us to define a suitable notion of neighbourhood.

1.4.10 Definition Let X be a set, let $\circ$ be a d-membership relation on X and let $\mathcal{U}_x = \emptyset$ be a collection of subsets of X for each $x \in X$. We call $(\mathcal{U}_x, \lessdot)$ a *d-neighbourhood system* (*d-nbhood system*) *for* x if it satisfies the following conditions.

(Ni) If $U \in \mathcal{U}_x$, then $x \lessdot U$.

(Nii) If $U, V \in \mathcal{U}_x$, then $U \cap V \in \mathcal{U}_x$.

(Niii) If $U \in \mathcal{U}_x$, then there is a $V \quad U$ with $V \in \mathcal{U}_x$ such that for all $y \lessdot V$ we have $U \in \mathcal{U}_y$.

(Niv) If $U \in \mathcal{U}_x$ and $U \quad V$, then $V \in \mathcal{U}_x$.

Each $U \in \mathcal{U}_x$ is called a *d-neighbourhood* (*d-nbhood*) of x. Finally, let X be a set, let $\circ$ be a d-membership relation on X and, for each $x \in X$, let $(\mathcal{U}_x, \lessdot)$ be a d-nbhood system for x. Then $(X, \mathcal{U}, \lessdot)$ (or simply X) is called a *d-topological space*, where $\mathcal{U} = \{\mathcal{U}_x \mid x \in X\}$.

Note that points may have empty d-nbhoods and that Definition 1.4.10 is exactly the definition of a topological neighbourhood system if $\circ$ is the membership relation $\in$.

Proposition 1.4.11, next, shows that d-nbhood systems arise naturally from d-metrics.

1.4.11 Proposition Let (X, ϱ) be a d-metric space. Define the d-membership relation $\circ$ as the relation $\{(x, A) \mid$ there exists $\varepsilon > 0$ for which $B_\varepsilon(x) \quad A\}$. For each $x \in X$, let $\mathcal{U}_x$ be the collection of all subsets A of X such that $x \lhd A$. Then $(\mathcal{U}_x, \lhd)$ is a d-nbhood system for x for each $x \in X$.

Proof: It is easy to see that $\circ$ is indeed a d-membership relation.
(Ni) is obvious. Note that we also have the reverse property: if $x \lhd U$, then $U \in \mathcal{U}_x$.
(Nii) If $x \lhd U, V$, then there are balls A, B with centre x such that $A \quad U$ and $B \quad V$. Without loss of generality let A be the smaller of the balls A and B. Then $A = A \cap B \quad U \cap V$.
(Niii) Let $U \in \mathcal{U}_x$, that is, $x \lhd U$. Then there is a ball B with centre x such that $B \quad U$ and $B \in \mathcal{U}_x$. Now let $y \lhd B$ be arbitrary. We have to show that $y \lhd U$. But $y \lhd B$ implies that there is a ball B' with centre y such that $y \lhd B' \quad B \quad U$. So $y \lhd U$.
(Niv) This is obvious since $x \lhd U \quad V$ implies $x \lhd V$. ■

We note that if (X, ϱ) is a metric space, then the above construction yields the usual topology associated with a metric.

The set of balls of a d-metric does not in general yield a conventional topology. In this respect, the axioms defining a dislocated metric are different from those defining a partial metric in [Mat92, Mat94], which are as follows.

1.4.12 Definition Let X be a set and let $p : X \times X \to \mathbb{R}^+$ be a function. We call p a *partial metric* on X if it satisfies the following axioms.

(Pi) For all $x, y \in X$, $x = y$ if and only if $p(x, x) = p(x, y) = p(y, y)$.

(Pii) For all $x, y \in X$, $p(x, x) \leq p(x, y)$.

(Piii) For all $x, y \in X$, $p(x, y) = p(y, x)$.

(Piv) For all $x, y, z \in X$, $p(x, z) \leq p(x, y) + p(y, z) - p(y, y)$.

A *weak partial metric* is a distance function satisfying conditions (Pi), (Piii) and (Piv) of Definition 1.4.12, i.e. condition (Pii) of *small self-distances* is not required. These spaces were studied e.g. in [EH98, Hec99, O'N95], and we note that [O'N95] works with partial metrics where negative distances are allowed. It is easy to see that any (weak) partial metric is a d-metric. Furthermore, the set of balls with respect to a (weak) partial metric does indeed yield a topology, and strong relationships between the topologies arising from partial metrics and topologies discussed in domain theory can be established. We refer the reader to

[Mat92, Mat94, EH98, Wac00] for a comprehensive discussion of these matters since our main concern here is with the more general notion of dislocated metric. We will not follow the lines mentioned in this paragraph since dislocated metrics will suffice for the purpose of our applications.

1.4.13 Proposition Any d-ultrametric satisfies (Pii), (Piii) and (Piv), but not necessarily (Pi).

Proof: Let (X, ϱ) be a d-ultrametric space and let $x, y, z \in X$.
(Pii) By the strong triangle inequality, we obtain $\varrho(x,x) \leq \max\{\varrho(x,y), \varrho(y,x)\}$ and by symmetry we obtain the desired inequality.
(Piii) follows from (Miii).
(Piv) By the strong triangle inequality, we obtain $\varrho(x,z) \leq \max\{\varrho(x,y), \varrho(y,z)\}$. Without loss of generality, we can assume that $\varrho(x,y) \geq \varrho(y,z)$. Since by (Pii) we have $\varrho(y,y) \leq \varrho(y,z)$, we obtain $\varrho(x,z) \leq \varrho(x,y) \leq \varrho(x,y) + \varrho(y,z) - \varrho(y,y)$.

Let X be a set and define ϱ on $X \times X$ to be identically 1. Then ϱ is a d-ultrametric on X which does not satisfy (Pi). ■

Convergence and Continuity

Once the notion of d-nbhood is defined, it is straightforward to adapt the notion of convergence to d-topological spaces.

1.4.14 Definition Let $(X, \mathcal{U}, \lessdot)$ be a d-topological space and let $x \in X$. A (topological) net (x_λ) *d-converges* to $x \in X$ if for each d-nbhood U of x we have that x_λ is *eventually in* U, that is, there exists some λ such that $x_\lambda \in U$ for each $\lambda > \lambda$.

Note that if for some $x \in X$ we have $\emptyset \in \mathcal{U}_x$, then the constant sequence (x) does not d-converge. In fact, if $\emptyset \in \mathcal{U}_x$, then no net in X d-converges to x. Note also that the notion of convergence obtained in Definition 1.4.14 is a natural generalization of convergence with respect to a d-metric, and we investigate this next.

1.4.15 Proposition Let (X, ϱ) be a d-metric space and let $(X, \mathcal{U}, \lessdot)$ be the d-topological space obtained from it via the construction in Proposition 1.4.11. Let (x_n) be a sequence in X. Then (x_n) converges in ϱ if and only if (x_n) d-converges in $(X, \mathcal{U}, \lessdot)$.

Proof: Let (x_n) be convergent in ϱ to some $x \in X$, so that $\varrho(x_n, x) \to 0$ as $n \to$, and let U be a d-nbhood of x. Then there exists $\varepsilon > 0$ such that $B_\varepsilon(x)$ U. Since $\varrho(x_n, x) \to 0$, there exists n such that $x_n \in B_\varepsilon(x)$ U for all $n > n$ and hence (x_n) d-converges to x.

Conversely, let (x_n) be d-convergent to some $x \in X$, that is, for each d-nbhood U of x there exists n such that $x_n \in U$ for each $n > n$. For each $\varepsilon > 0$, $B_\varepsilon(x)$ is a d-nbhood of x. Since ε can be chosen arbitrarily small, we must have $\varrho(x_n, x) \to 0$ as $n \to \infty$, as required. ■

We proceed with defining continuity on d-topological spaces.

1.4.16 Definition Let X and Y be d-topological spaces and let $f : X \to Y$ be a function. Then f is *d-continuous* at $x_0 \in X$ if for each d-nbhood V of $f(x_0)$ in Y there is a d-nbhood U of x_0 in X such that $f(U) \subseteq V$. We say f is *d-continuous* on X if f is d-continuous at each $x_0 \in X$.

The following theorem shows that the notion of d-convergence can be characterized via nets, by analogy with conventional topology.

1.4.17 Theorem Let X and Y be d-topological spaces and let $f : X \to Y$ be a function. Then f is d-continuous if and only if for each net (x_λ) in X which d-converges to some $x_0 \in X$, $(f(x_\lambda))$ is a net in Y which d-converges to $f(x_0) \in Y$.

Proof: Let f be d-continuous at x_0 and let x_λ be a net which d-converges to x_0. Let V be a d-nbhood of $f(x_0)$. Then there exists a d-nbhood U of x_0 such that $f(U) \subseteq V$. Since x_λ is eventually in U, we obtain that $f(x_\lambda)$ is eventually in V, and hence $f(x_\lambda)$ d-converges to $f(x_0)$.

Conversely, if f is not d-continuous at x_0, then for some d-nbhood V of $f(x_0)$ and for all $U \in \mathcal{U}_{x_0}$ we have $f(U) \not\subseteq V$. Thus for each $U \in \mathcal{U}_{x_0}$ there is an $x_U \in U$ with $f(x_U) \in V$. Then (x_U) is a net in X which d-converges to x_0 whilst $f(x_U)$ does not d-converge to $f(x_0)$. ■

We have generalized convergence from d-metrics to d-topologies. However, we still lack a notion of continuity in terms of d-metrics. We will investigate this next, and this will enable us to give a proof of the Matthews theorem 1.4.6 which is analogous to the standard proof of the Banach contraction mapping theorem.

1.4.18 Proposition Let (X, ϱ) and (Y, ϱ') be d-metric spaces, let $f : X \to Y$ be a function and let $(X, \mathcal{U}, \lessdot)$ and $(Y, \mathcal{V}, \lessdot')$ be the d-topological spaces obtained from (X, ϱ), respectively (Y, ϱ'), via the construction in Proposition 1.4.11. Then f is d-continuous at $x_0 \in X$ if and only if for each $\varepsilon > 0$ there exists a $\delta > 0$ such that $f(B_\delta(x_0)) \subseteq B_\varepsilon(f(x_0))$.

Proof: Let f be d-continuous at $x_0 \in X$ and let $\varepsilon > 0$. Then $B_\varepsilon(f(x_0))$ is a d-nbhood of $f(x_0)$. By definition of d-continuity, there exists a d-nbhood U of x_0 with $f(U) \subseteq B_\varepsilon(f(x_0))$. But since U is a d-nbhood of x_0, there exists a ball $B_\delta(x_0) \subseteq U$ and therefore $f(B_\delta(x_0)) \subseteq f(U) \subseteq B_\varepsilon(f(x_0))$.

Conversely, assume that the ε-δ-condition on f holds and let V be a d-nbhood of $f(x_0)$. Then there exists $\varepsilon > 0$ with $B_\varepsilon(f(x_0)) \subseteq V$ and $\delta > 0$ with $f(B_\delta(x_0)) \subseteq B_\varepsilon(f(x_0)) \subseteq V$. Since $B_\delta(x_0)$ is a d-nbhood of x_0 we obtain d-continuity of f. ■

1.4.19 Proposition Let (X, ϱ) be a d-metric space, let $f : X \to X$ be a contraction with contractivity factor λ and let $(X, \mathcal{U}, \lessdot)$ be the d-topological space obtained from (X, ϱ) via the construction in Proposition 1.4.11. Then f is d-continuous.

Proof: Let $x \ \in X$ and let $\varepsilon > 0$ be arbitrarily chosen. For $\delta = \frac{\varepsilon}{\lambda+1}$, we obtain $d(f(x), f(x \ )) \leq \lambda d(x, x \ ) \leq \lambda \frac{\varepsilon}{\lambda+1} \quad \varepsilon$ for all $x \in B_\delta(x \ )$, and therefore $f\,(B_\delta(x \ )) \quad B_\varepsilon(f(x \ ))$ as required. ∎

Proof of Theorem 1.4.6: With our preparations, the proof follows the proof of the Banach contraction mapping theorem on metric spaces, and we only sketch the details here.

Let $x \in X$ be arbitrarily chosen. Then the sequence $(f^n(x))_{n\in\mathbb{N}}$ is a Cauchy sequence and converges in (X, ϱ) to some point y. Since f is a contraction, it is also d-continuous by Proposition 1.4.19 from which we obtain $y = \lim f^n(x) = f(\lim f^{n-1}(x)) = f(y)$ by Theorem 1.4.17. Uniqueness follows since if z is a fixed point of f, then $\varrho(x, z) = \varrho(f(x), f(z)) \leq \lambda\varrho(x, z)$ and therefore $\varrho(x, z) = 0$, and hence $x = z$ by (Mii). ∎

1.5 Dislocated Generalized Ultrametrics

The following theorem gives a partial unification of the Matthews theorem 1.4.6 and the Prieß-Crampe and Ribenboim theorem 1.3.4. The proof of the latter theorem given in [PCR93] in fact carries over directly to our more general setting of d-gums.

1.5.1 Theorem Let (X, ϱ, Γ) be a spherically complete d-gum space and let $f : X \to X$ be non-expanding and strictly contracting on orbits. Then f has a fixed point. If f is strictly contracting on X, then the fixed point is unique.

Proof: Assume that f has no fixed point. Then for all $x \in X$ we have $\varrho(x, f(x)) = 0$. We define the set $\mathcal{B}$ by $\mathcal{B} = \{B_{\varrho(x,f(x))}(x) \mid x \in X\}$, and note that each ball in this set is non-empty. We also note that $B_{\varrho(x,f(x))}(x) = B_{\varrho(x,f(x))}(f(x))$ by Lemma 1.3.3. Now let $\mathcal{C}$ be a maximal chain in $\mathcal{B}$. Since X is spherically complete, there exists $z \in \quad \mathcal{C}$. We show that $B_{\varrho(z,f(z))}(z) \quad B_{\varrho(x,f(x))}$ for all $x \in X$ and hence, by maximality, that $B_{\varrho(z,f(z))}(z)$ is the smallest ball in the chain. Let $B_{\varrho(x,f(x))}(x) \in \mathcal{C}$. Since $z \in B_{\varrho(x,f(x))}(x)$, and noting our earlier observation that $B_{\varrho(x,f(x))}(x) = B_{\varrho(x,f(x))}(f(x))$ for all x, we get $\varrho(z, x) \leq \varrho(x, f(x))$ and $\varrho(z, f(x)) \leq \varrho(x, f(x))$. By non-expansiveness of f, we get $\varrho(f(z), f(x)) \leq \varrho(z, x) \leq \varrho(x, f(x))$. It follows by (Uiv) that $\varrho(z, f(z)) \leq \varrho(x, f(x))$ and therefore that $B_{\varrho(z,f(z))}(z) \quad B_{\varrho(x,f(x))}(x)$ by Lemma 1.3.3 for all $x \in X$, since x was chosen arbitrarily. Now, since f is strictly contracting on orbits, $\varrho(f(z), f^2(z)) \quad \varrho(z, f(z))$, and therefore $z \in B_{\varrho(f(z),f^2(z))}(f(z)) \subset B_{\varrho(z,f(z))}(f(z))$. By Lemma 1.3.3, this is equivalent to $B_{\varrho(f(z),f^2(z))}(f(z)) \subset B_{\varrho(z,f(z))}(z)$, 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 $\varrho(x, y) = \varrho(f(x), f(y)) \quad \varrho(x, y)$ which is impossible. So the fixed point of f is unique in this case. ∎

We next give a constructive proof of a special case of Theorem 1.5.1.

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

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

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

It is straightforward to cast the above observations into a transfinite induction argument, and we obtain the following construction:

Choose $x \in X$ arbitrarily. For each ordinal $\alpha \leq \gamma$, we define $f^\alpha(x)$ as follows. If α is a successor ordinal, then $f^\alpha(x) = f(f^{\alpha-1}(x))$ as usual. If α is a limit ordinal, then we choose $f^\alpha(x)$ as some x_α which has the property that $d(x_\alpha, f^\beta(x)) \leq 2^{-\beta}$, and the existence of such an x_α is guaranteed by spherical completeness of X.

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

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

Another alternative proof of this theorem will be given at the end of Section 3.4.

1.6 Quasimetrics

Quasimetrics are a convenient way of reconciling metric and order structures. We give the relevant definitions in order to state the Rutten-Smyth theorem 1.6.3, in the form in which it appears in [Rut96]. A more general version was given in [Smy87] on quasi-uniformities.

1.6.1 Definition A sequence (x_n) in a quasimetric space (X, d) is a *(forward) Cauchy sequence* if, for all $\varepsilon > 0$, there exists $n \ \in \mathbb{N}$ such that for all $n \geq m \geq n$

we have $d(x_m, x_n) \quad \varepsilon$. A Cauchy sequence (x_n) *converges* to $x \in X$ if, for all $y \in X$, $d(x, y) = \lim d(x_n, y)$. Finally, X is called *CS-complete* if every Cauchy sequence in X converges.

Note that limits of Cauchy sequences in quasimetric spaces are unique. Given a quasimetric space (X, d), d induces a partial order $\leq_d$ on X by setting $x \leq_d y$ if and only if $d(x, y) = 0$. If (X, d) is a quasimetric space, then (X, d^*) is a metric space, where $d^*(x, y) = \max\{d(x, y), d(y, x)\}$.

1.6.2 Definition Let X be a quasimetric space. A function $f : X \to X$ is called

(1) *CS-continuous* if, for all Cauchy sequences (x_n) in X with $\lim x_n = x$, $(f(x_n))$ is a Cauchy sequence 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 $d(f(x), f(y)) \leq c \quad d(x, y)$ for all $x, y \in X$.

Contractive mappings are not necessarily CS-continuous as was pointed out in [Rut96], where also a proof of the following theorem can be found.

1.6.3 Theorem (Rutten-Smyth theorem) Let (X, d) be a CS-complete quasimetric space 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 Cauchy sequence $(f^n(x))$, where in (1) x is the given point, and in (2) x can be chosen arbitrarily.

Let $(X, \leq)$ be a partially ordered set. Define a function $d_{\leq} : X \times X \to \mathbb{R}^+$ by

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

Then it is easily checked that $(X, d_{\leq})$ is a quasi-ultrametric space, and $d_{\leq}$ is called the *discrete quasimetric* on X. Note that $\leq_{d_{\leq}}$ and $\leq$ coincide for a given partial order $\leq$.

By virtue of this definition and the definition of $\leq_d$ for a given quasimetric d, Part (1) of Theorem 1.6.3 generalizes the Kleene theorem 1.1.3. Part (2) generalizes the Banach contraction mapping theorem 1.2.2, cf. also [Rut96, Smy87] and Proposition 2.4.4.

space	name of theorem	reference number	symbol
ω-cpo	Kleene	1.1.3	K
chain-complete partial order	Knaster-Tarski	1.1.7	KT
complete metric	Banach	1.2.2	B
compact metric	—	1.2.3	cp
gum	Prieß-Crampe and Ribenboim	1.3.4	PCR
d-metric	Matthews	1.4.6	M
d-gum	—	1.5.1	dPCR
quasimetric	Rutten-Smyth	1.6.3	RS

Table 1.3: Summary of single-valued fixed-point theorems.

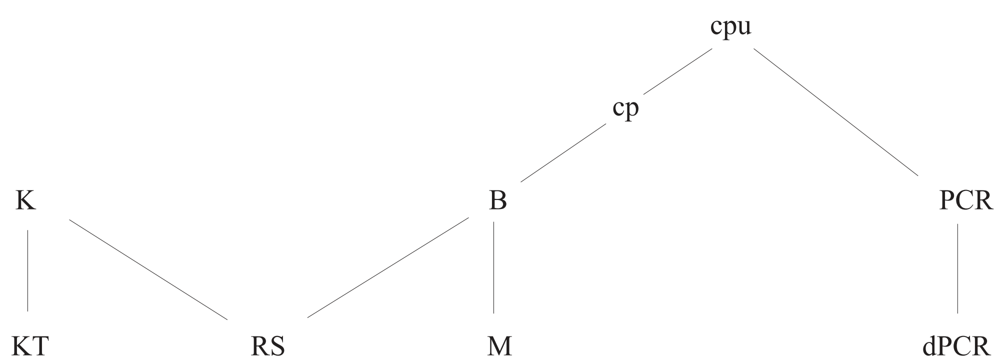

Figure 1.1: Dependencies between fixed-point theorems from Chapter 1. If a theorem is depicted lower in the diagram, this means that it is more general. See Table 1.3 for the abbreviations.

1.7 Summary and Further Work

We have presented a number of theorems on different order structures and generalized metrics, which are collected in Table 1.3.

The dependencies between these theorems are depicted in Figure 1.1, where the letters abbreviate the theorems as listed in Table 1.3. The abbreviation "cpu" stands for the fact that strictly contracting functions on compact ultrametric spaces have unique fixed points, which is an easy corollary of Theorem 1.2.3.

We note that the Prieß-Crampe and Ribenboim theorem 1.3.4 can be proven using the Knaster-Tarski theorem 1.1.7, analogous to a proof in [EH98] of the Banach contraction mapping theorem 1.2.2 from the Kleene theorem 1.1.3, see Section 3.3. Also, the dislocated Prieß-Crampe and Ribenboim theorem 1.5.1, respectively the Matthews theorem, can be proven using the non-dislocated version, i.e. the Prieß-Crampe and Ribenboim theorem 1.3.4, respectively the Banach contraction mapping theorem 1.2.2, see Sections 3.4 and 3.1, respectively.

We list a number of questions arising from our results.

Question 1.1 Is there a reasonable notion of *d-open set* corresponding to the notions of d-neighbourhood, d-convergence and d-continuity as in Section 1.4?

Question 1.2 What are necessary and sufficient conditions such that a spherically complete gum is compact?

Question 1.3 Is there a quasimetric version of the Prieß-Crampe and Ribenboim theorem 1.3.4?

Question 1.4 Which of the theorems in Figure 1.1 allow for common generalizations?

Chapter 2

Fixed-point Theorems for Multivalued Mappings

We briefly present fixed-point theorems for multivalued mappings on partial orders and generalized metrics, and study some of the relationships between them. It turns out that many fixed-point theorems from Chapter 1 can be carried over to a multivalued setting. In Section 2.1, we carry over the Knaster-Tarski theorem 1.1.7. In Section 2.2, we present a multivalued version of the Banach contraction mapping theorem 1.2.2. Section 2.3 is concerned with multivalued variants of the Prieß-Crampe and Ribenboim theorem and Section 2.4 introduces a theorem for multivalued mappings on quasimetrics which reconciles the theorems on partial orders and metrics analogous to the Rutten-Smyth theorem 1.6.3.

2.1 Partial Orders

We review a multivalued version of the Knaster-Tarski theorem 1.1.7 due to [KM98]. A multivalued Kleene theorem will be presented in Section 2.4.

2.1.1 Definition Let $T : X \to 2^X$ be a multivalued mapping defined on X. An *orbit* of T is a net $(x_i)_{i\in\alpha}$ in X, where α denotes an ordinal, such that $x_{i+1} \in T(x_i)$ for all $i \in \alpha$. An orbit $(x_i)_{i\in\alpha}$ of T is called an *ω-orbit* if α is the first limit ordinal, ω. An orbit $(x_i)_{i\in\alpha}$ of T will be said to be *eventually constant* if there is a tail $(x_i)_{\beta\leq i}$ of $(x_i)_{i\in\alpha}$ which is *constant* in that $x_i = x_j$ for all $i, j \in \alpha$ satisfying $\beta \leq i, j$.

If $T : X \to 2^X$ is a multivalued mapping and x is a fixed point of T, then we obtain an orbit of T which is eventually constant by setting $x = x \ \ = x_1 = x_2 \ldots$. Conversely, suppose that $(x_i)_{i\in\alpha}$ is an orbit of T with the property that $x_{i+1} = x_i$ for all $i \in \alpha$ satisfying $\beta \leq i$, for some ordinal $\beta \in \alpha$. Then $x_\beta = x_{\beta+1} \in T(x_\beta)$ and we have a fixed point x_β of T. Thus, having a fixed point and having an orbit which is eventually constant are essentially equivalent conditions on T.

2.1.2 Definition A multivalued mapping T defined on a partially ordered set X will be said to be *monotonic* if, for all $x, y \in X$ satisfying $x \leq y$ and for all $a \in T(x)$, there exists $b \in T(y)$ such that $a \leq b$.

2.1.3 Definition An orbit $(x_i)_{i \in \alpha}$ of T is said to be *increasing* if we have $x_i \leq x_j$ for all $i, j \in \alpha$ satisfying $i \leq j$, and is said to be *eventually increasing* if some tail of the orbit is increasing. Finally, an increasing orbit $(x_i)_{i \in \alpha}$ of T is said to be *tight* if, for all limit ordinals $\beta \in \alpha$, we have $x_\beta = \sup\{x_i \mid i< \ \beta\}$.

Suppose that $(x_i)_{i \in \alpha}$ is an increasing orbit of T and that $\beta \in \alpha$ is a limit ordinal. Then $x_{\beta+1}$ is an element of $T(x_\beta)$ such that $x_i \leq x_{\beta+1}$ for all $i \quad \beta$, and of course $\sup\{x_i \mid i \quad \beta\} \leq x_\beta \leq x_{\beta+1}$ if the supremum exists. In particular, any increasing orbit $(x_i)_{i \in \alpha}$ which is tight (if such exists) must satisfy the following condition:

$$\text{For any limit ordinal } \beta, \text{ there exists } x \ (= x_{\beta+1}) \in T(\sup\{x_i \mid i< \ \beta\})$$
$$\text{such that } \sup\{x_i \mid i \quad \beta\} \leq x. \tag{2.1}$$

This condition is a slight variant of a condition which was identified in [KM98] as a sufficient condition for the existence of fixed points of monotonic multivalued mappings. In fact, the following result was established in [KM98], except that it was formulated for decreasing orbits and infima and we have chosen to work with the dual notions instead, to maintain consistency.

2.1.4 Theorem (Knaster-Tarski multivalued) Let X be a complete partial order and let $T : X \to 2^X$ be a multivalued mapping which is non-empty, monotonic and satisfies (2.1). Then T has a fixed point.

We omit details of the proof of this result except to observe that, starting with the bottom element $x \quad = \perp$ of X, the condition (2.1) permits the construction, transfinitely, of a tight orbit (x_i) of T. Since this can be carried out for ordinals whose underlying cardinal is greater than that of X, we are forced to conclude that (x_i) is eventually constant and therefore that T has a fixed point.

Noting that $\sup\{x_i \mid i< \ \beta\} = \sup\{x_{i+1} \mid i \quad \beta\}$, one can view (2.1) schematically as the statement "$\sup\{T(x_i) \mid i \quad \beta\} \leq T(\sup\{x_i \mid i \quad \beta\})$" and it can therefore be thought of as a rather natural, weak continuity condition on T which is automatically satisfied by any monotonic single-valued mapping T on a cpo. The question of when the orbit constructed in the previous paragraph becomes constant in ω steps as in the single-valued Kleene theorem 1.1.3 is a question of continuity and will be taken up in Section 2.4.

Theorem 2.1.4 was established in [KM98] in order to show the existence of (consistent) answer sets for a class of disjunctive programs called signed programs, see Section 7.3. At the end of Section 7.3, we will give examples which show that it sometimes is necessary to work transfinitely in practice, a point which justifies the name "Knaster-Tarski theorem" applied to Theorem 2.1.4.

Thus, to summarize, monotonicity of T together with (2.1) appears to give, for multivalued mappings, an exact analogue of the fixed-point theory for monotonic single-valued mappings due to Knaster-Tarski. Moreover, there are applications to the semantics of disjunctive programs which parallel those made in the standard, non-disjunctive case.

2.2 Metrics

We present a result due to [KKM93] which is a multivalued version of the Banach contraction mapping theorem 1.2.2.

2.2.1 Definition Let (X, d) be a metric space. A multivalued mapping $T : X \to 2^X$ is called a *contraction* if there exists a real number $k< 1$ such that for every $x \in X$, for every $y \in X$, and for all $a \in T(x)$, there exists $b \in T(y)$ such that $d(a, b) \leq kd(x, y)$.

The following result is taken from [KKM93]. An alternative proof will be given in Section 2.4.

2.2.2 Theorem (Banach multivalued) Assume that X is a complete metric space, and that T is a multivalued contraction on X such that, for every $x \in X$, the set $T(x)$ is closed and non-empty. Then T has a fixed point.

This theorem was established with a specific objective in view, namely, to show the existence of answer sets for disjunctive logic programs which are countably stratified [KKM93].

2.3 Generalized Ultrametrics

We present multivalued versions of the Prieß-Crampe and Ribenboim theorem 1.3.4.

2.3.1 Definition Let (X, d, Γ) be a generalized ultrametric space (so that Γ is a partially ordered set). A multivalued mapping T on X is called *strictly contracting*, respectively, *non-expanding* if, for all $x, y \in X$ with $x = y$ and for every $a \in T(x)$, there exists an element $b \in T(y)$ such that $d(a, b)$ $d(x, y)$, respectively, $d(a, b) \leq d(x, y)$.

The mapping T is called *strictly contracting on orbits*, if for every $x \in X$ and for every $a \in T(x)$ with $a = x$, there exists an element $b \in T(a)$ with $d(a, b)$ (a, x). $<d$

For $T : X \to 2^X$, let $\Pi_x = \{d(x, y) \mid y \in T(x)\}$ and, for a subset Δ Γ, denote by $\mathrm{Min}\,\Delta$ the set of all minimal elements of Δ.

The following theorem was proved in [PCR00c]. Although we know of no specific application of it, we believe it will prove to be useful by virtue of the general nature of the set Γ.

2.3.2 Theorem (Prieß-Cramps and Ribenboim multivalued) Let (X, d) be a spherically complete generalized ultrametric space. Let $T : X \to 2^X$ be non-empty, non-expanding and strictly contracting on orbits. Moreover, assume that for every $x \in X$, $\operatorname{Min}\Pi_x$ is finite and that every element of Π_x has a lower bound in $\operatorname{Min}\Pi_x$. Then T has a fixed point.

The following ideas were considered in [KKM93]. We show that the notions defined there basically coincide with those from generalized ultrametrics.

2.3.3 Definition A *semigroup* is a set V together with an associative binary operation $\quad : V \times V \to V$. If is also commutative, then the semigroup is called *commutative* or *Abelian*. A semigroup is called a *semigroup with* 0 if there exists an element $0 \in V$ such that $0 \quad u = u \qquad = u$ for all $u \in V$. +0

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

2.3.4 Definition Let V be an ordered Abelian semigroup with 0 and let X be an arbitrary set. A *g-metric* on X is a mapping $d : X \times X \to V$ which satisfies the following conditions for all $x, y, z \in X$.

1. $d(x, y) = 0$ if and only if $x = y$.
2. $d(x, y) = d(y, x)$.
3. $d(x, y) \leq d(x, z)+\ d(z, y)$.

A pair (X, d) consisting of a set X and a g-metric d on X is called a *g-metric space*.

In [KKM93], g-metrics were called generalized metrics, but we have changed the notation since the notion of generalized metric is used differently in this thesis. We will in fact not work with g-metrics in the sequel since the strongly related generalized ultrametrics will suffice for our purposes. We investigate this relationship next; the following definitions are again taken from [KKM93].

2.3.5 Definition Let V denote the set of all expressions of the type 0 or $2^{-\alpha}$, where α is a countable ordinal. An order is defined on V by: $0 \leq v$ for every $v \in V$, and $2^{-\alpha} \leq 2^{-\beta}$ if and only if $\beta \leq \alpha$. As a semigroup operation $u \quad v$, we will use the maximum $\max(u, v)$. It will be convenient to write $\frac{1}{2}2^{-\alpha} = 2^{-(\alpha+1)}$.

2.3.6 Definition Assume that α is either a countable ordinal or ω_1, the first uncountable ordinal, and that $\mathbf{v} = (v_\beta)_{\beta<\alpha}$ is a decreasing family of elements of V. Let X be a g-metric space, and let $(x_\beta)_{\beta<\alpha}$ be a family of elements of X.

(i) (x_β) is said to $\mathbf{v}$-*cluster* to $x \in X$ if, for all β, we have $d(x_\beta, x) \quad v_\beta$ whenever $\beta \quad$. $<\alpha$

(ii) (x_β) is said to be **v**-*Cauchy* if, for all β and γ, we have $d(x_\beta, x_\gamma)$ v_β whenever β . $<\gamma$ $<\alpha$

(iii) X is said to be *complete* if for every **v**, every **v**-Cauchy family **v**-clusters to some element in X.

(iv) A set A X will be called *complete* if for every **v**, whenever a **v**-Cauchy family consists of elements of A, it **v**-clusters to some element of A.

A strong relationship between the notion of completeness of g-metrics with the notion of trans-completeness, Definition 1.3.6, for generalized ultrametrics is obvious. We show that they coincide by showing equivalence between completeness for g-metrics and spherical completeness for generalized ultrametrics, cf. Proposition 1.3.8.

2.3.7 Definition A mapping $T : X \to 2^X$ is called a $\left(\frac{1}{2}\right)$-*contraction* if, for every $x \in X$, for every $y \in X$ and for every $a \in T(x)$, there exists $b \in T(y)$ such that $d(a,b) \leq \frac{1}{2} d(x,y)$.

The following theorem was proved in [KKM93].

2.3.8 Theorem Let X be a complete g-metric space, let T be a multivalued $\left(\frac{1}{2}\right)$-contraction on X such that $T(x)$ is not empty for some $x \in X$ (i.e. T is not identically empty), and suppose that for every $x \in X$ the set $T(x)$ is complete. Then T has a fixed point.

We present some results relating the results just given to the notion of spherical completeness we discussed earlier.

Let (X, d) be a g-metric space with respect to V as given in Definition 2.3.5. Then d is in fact a generalized ultrametric space and vice-versa.

2.3.9 Proposition Let (X, d) be a complete g-metric space with respect to V. Then X is spherically complete as an ultrametric space.

Proof: Let $\mathcal{B} = \left(B_{v_\beta}(x_\beta)\right)_{\beta<\alpha}$ be a decreasing chain of balls in X, and without loss of generality assume that it is strictly decreasing and that α is a limit ordinal. We have to show that $\mathcal{B} = \emptyset$. Let $\mathbf{v} = (v_\beta)_\beta$. Since $\mathcal{B}$ is a chain, it is easy to see that $(x_{\beta+1})_\beta$ is **v**-Cauchy and therefore, by completeness of X, $(x_{\beta+1})$ **v**-clusters to some $x \in X$. By definition, this means that $d(x_{\beta+1}, x)$ $_\beta$ and therefore that $x \in B_{v_\beta}(x_{\beta+1}) = B_{v_\beta}(x_\beta)$ for all β. Thus, $x \in$ $\mathcal{B}$. ■

2.3.10 Proposition Let (X, d, V) be a spherically complete generalized ultrametric space. Then X is complete as a g-metric space.

Proof: Let $\mathbf{v}=(v_\beta)$ be a decreasing family of elements of V which is, without loss of generality, strictly decreasing, and let (x_β) be $\mathbf{v}$-Cauchy. For $v \in \mathbf{v}$, e.g. $v = 2^{-\alpha}$, let v' denote $2^{-(\alpha+1)}$. Then $\mathcal{B} = \left(B_{v'_\beta}(x_\beta)\right)_\beta$ is a decreasing chain of balls in X. By spherical completeness, it has non-empty intersection. Choose $x \in \quad \mathcal{B}$. Then for all β we obtain $d(x_\beta, x) \leq v'_\beta \quad {}_\beta$, i.e. (x_β) $\mathbf{v}$-clusters to x. ■

This means, by virtue of Theorem 2.3.2, that we can reformulate the assumptions in Theorem 2.3.8 and thereby obtain the following theorem which in fact is a special case of [PCR00c, (3.4)].

2.3.11 Theorem Let X be a spherically complete generalized ultrametric space (with respect to V) and let T be multivalued, non-empty and strictly contracting on X and s.t. $T(x)$ is spherically complete for all $x \in X$. Then T has a fixed point.

2.4 Quasimetrics

We study a multivalued version of the Rutten-Smyth theorem 1.6.3, which will lead to a multivalued version of the Kleene theorem 1.1.3.

2.4.1 Definition Let (X, d) be a quasimetric space. A multivalued mapping $T : X \to 2^X$ is called a *contraction* if there exists a λ with $0 \leq \lambda < 1$ such that, for all $x, y \in X$ and for all $a \in T(x)$, there exists $b \in T(y)$ satisfying $d(a,b) \leq \lambda d(x,y)$. We say that T is *non-expanding* if, for all $x, y \in X$ and for all $a \in T(x)$, there exists $b \in T(y)$ satisfying $d(a,b) \leq d(x,y)$.

These definitions are clearly extensions of well-known definitions made for single-valued mappings, and indeed collapse to them in the case that T is single-valued. An obvious and natural definition of continuity of T is the following: for every Cauchy sequence (x_n) in X with limit x and for every choice of $y_n \in T(x_n)$, we have that (y_n) is a Cauchy sequence and $\lim y_n \in T(x)$. In fact, the weaker definition following, which is implied by the one just given, suffices for our purposes and will be used throughout.

2.4.2 Definition Let $T : X \to 2^X$ be a multivalued mapping defined on a quasimetric space (X, d). We say that T is *continuous* if we have $\lim x_n \in T(\lim x_n)$ for every ω-orbit (x_n) of T which is a Cauchy sequence.

Again, this definition collapses to a natural one in the case that T is single-valued. In fact, if T is single-valued, it simply states the condition that $\lim T(x_n) = \lim x_{n+1} = \lim x_n = T(\lim x_n)$ for every ω-orbit which is a Cauchy sequence, which is a weaker condition than that of CS-continuity as in Definition 1.6.2(1).

Finally, if (X, d) is a quasimetric space, we define the associated partial order $\leq_d$ on X by $x \leq_d y$ if and only if $d(x,y) = 0$, cf. Section 1.6.

The main result of this section is the following theorem, which generalizes the Rutten-Smyth theorem 1.6.3.

2.4.3 Theorem (Rutten-Smyth multivalued) Let (X, d) be a CS-complete quasimetric space and let $T : X \to 2^X$ denote a non-empty and continuous multivalued mapping on X. Then T has a fixed point if either of the following two conditions holds:

(a) T is a contraction.

(b) T is non-expanding and there is $x \in X$ and $x_1 \in T(x)$ such that $d(x, x_1) = 0$ i.e. $x \leq_d x_1$.

Proof: (a) Let $x \in X$. Since $T(x) = \emptyset$, we can choose $x_1 \in T(x)$. Since T is a contraction, there is $x_2 \in T(x_1)$ such that $d(x_1, x_2) \leq kd(x, x_1)$. Applying this argument repeatedly, we obtain a sequence (x_n) such that for all $n \geq 0$ we have $x_{n+1} \in T(x_n)$ and $d(x_{n+1}, x_{n+2}) \leq kd(x_n, x_{n+1})$. Thus, (x_n) is an ω-orbit. Using the triangle inequality, we obtain

$$d(x_n, x_{n+m}) \leq \sum_{i=0}^{m-1} d(x_{n+i}, x_{n+i+1}) \leq \sum_{i=0}^{m-1} k^{n+i} d(x, x_1) \leq \frac{k^n}{1-k} d(x, x_1).$$

Thus, (x_n) is a (forward) Cauchy sequence in X and therefore is an ω-orbit of T which is Cauchy. Since X is complete, (x_n) has a limit x. Now, by continuity of T, we obtain $x \in T(x)$ and x is a fixed point of T, as required.

(b) Let $x \in X$ and $x_1 \in T(x)$ satisfy $d(x, x_1) = 0$. Since T is non-expanding, there is $x_2 \in T(x_1)$ with $d(x_1, x_2) \leq d(x, x_1) = 0$. Inductively, we obtain a sequence (x_n) such that $x_{n+1} \in T(x_n)$ and $d(x_n, x_{n+k}) \leq \sum_{i=0}^{k-1} d(x_{n+i}, x_{n+i+1}) = 0$. Hence, (x_n) is an orbit of T which is forward Cauchy and therefore has a limit x. By continuity of T again, we see that x is a fixed point of T. ■

The proof given here of Part (a) of Theorem 2.4.3 is, up to the last step, exactly the same as the first half of the proof of the multivalued Banach contraction mapping theorem 2.2.2 established in [KKM93], except that we are working with a quasimetric rather than with a metric and therefore care needs to be taken that no use is made of symmetry. On the other hand, the proof we give next of Theorem 2.2.2, which roughly corresponds to the second half of the proof given in [KKM93], is shorter and technically somewhat simpler than the proof given in [KKM93].

We show next that Theorem 2.4.3 includes both the multivalued Banach contraction mapping theorem of [KKM93] just mentioned, and also a natural extension of the Kleene theorem 1.1.3 to multivalued mappings, see Theorem 2.4.6. As stated earlier, this unification is in direct analogy with the single-valued case.

Proof of Theorem 2.2.2 We show that the condition that $T(x)$ is closed for every x together with that of T being a contraction implies that T is continuous, and the result then follows from Part (a) of Theorem 2.4.3.

First note that (X, d) being a complete metric space means that (X, d) is complete as a quasimetric space, and obviously T satisfies (a) of Theorem 2.4.3. Now suppose that (x_n) is an orbit of T which is a forward Cauchy sequence and hence a Cauchy sequence; we want to show that $x \ \in T(x \ )$, where x is the limit of (x_n).

Since T is a contraction, for every n there exists $y_n \in T(x \ )$ such that $d(x_{n+1}, y_n) \leq kd(x_n, x \ )$. Therefore, $d(y_n, x \ ) \leq d(y_n, x_{n+1}) \quad d(x_{n+1}, x \ ) \leq kd(x_n, x \ )+ \ d(x_{n+1}, x \ )$. Hence, we have $y_n \to x$. But each $y_n \in T(x \ )$, and $T(x)$ is closed for every x. Consequently, the limit x of the sequence y_n also belongs to $T(x \ )$. So, $x \ \in T(x \ )$, and it follows that T is continuous as required. ■

We next turn our attention to demonstrating that Theorem 2.4.3 contains a version of the Kleene theorem for multivalued mappings. It will be necessary to make some preliminary observations, as follows, concerning partially ordered sets and the quasimetrics they carry. We refer to [Rut96] for these results.

2.4.4 Proposition Let $(X, \leq)$ be a partial order and let (X, d) denote the associated quasimetric space, i.e. $d = d_{\leq}$ as in Section 1.6. Then the following hold.

(i) A non-empty multivalued mapping $T : X \to 2^X$ is monotonic if and only if it is non-expanding.

(ii) A sequence (x_n) in X is eventually increasing in $(X, \leq)$ if and only if it is a Cauchy sequence in (X, d).

(iii) The partially ordered set $(X, \leq)$ is ω-complete if and only if (X, d) is complete as a quasimetric space. Furthermore, in the presence of either form of completeness, the limit of any Cauchy sequence is the least upper bound of any increasing tail of the sequence.

Notice that neither Part (iii) of this result nor the next definition assumes the presence of a bottom element.

2.4.5 Definition Let the partial order $(X, \leq)$ be ω-complete and let $T : X \to 2^X$ be a non-empty multivalued mapping on X. We say that T is *ω-continuous* if T is monotonic and, for any ω-orbit (x_n) of T which is eventually increasing, we have $\sup(x_n) \in T(\sup(x_n))$, where the supremum is taken over any increasing tail of (x_n).

We obtain finally the following Kleene theorem for multivalued mappings as an easy corollary of our Theorem 2.4.3. Some of its applications will be discussed in Section 7.3.

2.4.6 Theorem (Kleene multivalued) Let $(X, \leq)$ be an ω-complete partial order (with bottom element) and let $T : X \to 2^X$ be a non-empty, ω-continuous multivalued mapping on X. Then T has a fixed point.

space	name of theorem	reference number
ω-cpo	Kleene multivalued	2.4.6
cpo	Knaster-Tarski multivalued	2.1.4
complete metric	Banach multivalued	2.2.2
gum	Prieß-Crampe and Ribenboim multivalued	2.3.2
quasimetric	Rutten-Smyth multivalued	2.4.3

Table 2.1: Summary of multivalued fixed-point theorems.

Proof: Since $(X, \leq)$ is ω-complete, the associated quasimetric space (X, d) (i.e. $d = d_{\leq}$ as in Section 1.6) is complete by Proposition 2.4.4. Furthermore, T is monotonic, since it is ω-continuous, and is therefore non-expanding by Proposition 2.4.4 again. On taking $x\ = \perp$ and $x_1 \in T(x\ )$ arbitrarily, we have x and x_1 satisfying $d(x\ , x_1) = 0$. The result will therefore follow from Part (b) of Theorem 2.4.3 as soon as we have established that T is continuous in the sense of Definition 2.4.2.

Let (x_n) be any ω-orbit of T which is a Cauchy sequence. Then (x_n) is eventually increasing and, by ω-continuity of T, we have $\sup(x_n) \in T(\sup(x_n))$, where the supremum is taken over any increasing tail of (x_n). In other words, we have $\lim x_n \in T(\lim x_n)$ and hence we have the continuity of T that we require. ■

The Kleene theorem for single-valued mappings T asserts that the fixed point produced by the usual proof is the least fixed point of T. This assertion does not immediately carry over to the case of multivalued mappings T without additional assumptions. One such simple, though rather strong, condition is the following: for each $x \in X$, assume that $T(x)$ has a least element M_x and that $M_x \leq M_y$ whenever $x \leq y$. To see that this suffices, suppose that x is any fixed point of T, and construct the orbit (x_n) of T by setting $x\ = \perp$ and $x_{n+1} = M_{x_n}$ for each n. Then (x_n) converges to a fixed point $\overline{x}$. Noting that $\perp \leq x$ and that $M_x \leq x$, we see that $x_n \leq x$ for all n. Hence, $\overline{x} \leq x$.

2.5 Summary and Further Work

We summarize the fixed-point theorems presented in this chapter in Table 2.1, and note that these theorems have corresponding versions in the single-valued case which have been carried over. The obvious task of carrying over further single-valued fixed-point theorems along the same lines remains and should pose no particular difficulties.

We note that in the applications in Part II of the thesis, all gums will always have some ordinal, in reverse order, as distance set as in Definition 2.3.5, see also Sections 3.2 and 3.3. This is caused by the fact that the gums arising in our applications are derived from level mappings which are themselves mappings into ordinals.

We will employ multivalued mappings in the context of disjunctive logic programs in Section 7.3, where multivalued mappings naturally arise as semantic operators. In [ZR97a, ZR97b, ZR98], the authors avoid using multivalued mappings in the same context by using operators on powerdomains instead. And indeed, the monotonicity notions used in this chapter correspond to powerdomain constructions, more specifically to the Hoare powerdomain [SHLG94], which is an alternative to the Smyth powerdomain employed in [ZR98]. Details of these relationships remain to be worked out.

Chapter 3

Conversions between Spaces

We study relationships between the different spaces from Chapters 1 and 2. In particular, we will focus on the representation of some of the spaces by others, which will in some cases lead to alternative proofs for the respective fixed-point theorems.

In Section 3.1, we will establish relationships between conventional metrics and dislocated metrics. We will obtain several methods of obtaining dislocated metrics from metrics, some of which will be applied in Part II of the thesis, and we will show how the Matthews theorem 1.4.6 can be derived from the Banach contraction mapping theorem 1.2.2. In Section 3.2, we will see how Scott-Ershov domains can be cast into generalized ultrametric spaces, which will also be applied in Part II of the thesis. In Section 3.3 we will cast generalized ultrametric spaces into domains and derive another alternative proof of the Prieß-Crampe and Ribenboim theorem. Finally, in Section 3.4, we will study relationships between gums and d-gums analogous to Section 3.1.

We would like to note that quasimetrics are strongly related to partial orders, and we refer to [Smy87, Smy91, BvBR96, Rut96] for these matters since we will not make any specific use of these relationships in the sequel.

3.1 Metrics and Dislocated Metrics

In this section, we will investigate relationships between conventional metrics and d-metrics. First note that if f is a contraction with contractivity factor λ on a d-metric X, we have $\varrho(f(x), f(x)) \leq \lambda\varrho(x,x)$ for all $x \in X$. Since the requirement $\varrho(x,x) = 0$ for all $x \in X$ renders a d-metric to be a metric, we are interested in understanding the function $u_\varrho : X \to \mathbb{R}$ defined by $u_\varrho(x) = \varrho(x,x)$.

3.1.1 Definition Let (X,ϱ) be a d-metric space. The function $u_\varrho : X \to \mathbb{R} : x \to \varrho(x,x)$ is called the *dislocation function* of ϱ.

Depending on the context, dislocation functions are sometimes called *weight functions*, e.g. in [Mat94, Wac00].

3.1.2 Lemma Let (X, ϱ) be a d-metric space. Then $u_\varrho : X \to \mathbb{R}$ is d-continuous.

Proof: Recalling the observations following Definition 1.4.14, let $x \in X$ and let (x_λ) be a net in X which d-converges to x, that is, for each $\varepsilon > 0$ there exist λ such that $\varrho(x_\lambda, x)$ for all $\lambda > \lambda$. Since $u_\varrho(x_\lambda) = \varrho(x_\lambda, x_\lambda) \leq 2\varrho(x_\lambda, x)$ for all $<\varepsilon$ λ, we obtain $u_\varrho(x_\lambda) \to 0$ for increasing λ. It remains to show that $u_\varrho(x) = 0$, and this follows from $u_\varrho(x) = \varrho(x, x) \leq 2\varrho(x_\lambda, x)$, since the latter term tends to 0 for increasing λ. ■

The following is a general result which shows how d-metrics can be obtained from conventional metrics.

3.1.3 Proposition Let (X, d) be a metric space, let $u : X \to \mathbb{R}^+$ be a function and let $T : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ be a symmetric operator which satisfies the triangle inequality. Then (X, ϱ) with

$$\varrho(x, y) = d(x, y) + T(u(x), u(y))$$

is a d-metric space and $u_\varrho(x) = T(u(x), u(x))$ for all $x \in X$. In particular, if $T(x, x) = x$ for all $x \in \mathbb{R}^+$, then $u_\varrho \equiv u$.

Proof: (Mii) If $\varrho(x, y) = 0$, then $d(x, y) + T(u(x), u(y)) = 0$. Hence $d(x, y) = 0$ and $x = y$.
(Miii) Obvious by symmetry of d and T.
(Miv) Obvious since d and T satisfy the triangle inequality. ■

Completeness also carries over if some continuity conditions are imposed.

3.1.4 Proposition Using the notation of Proposition 3.1.3, let u be continuous as a function from (X, d) to $\mathbb{R}^+$ (endowed with the usual topology), and let T be continuous as a function from the topological product space $(\mathbb{R}^+)^2$ to $\mathbb{R}^+$, satisfying the additional property $T(x, x) = x$ for all x. If (X, d) is a complete metric space, then (X, ϱ) is a complete d-metric space.

Proof: Let (x_n) be a Cauchy sequence in (X, ϱ). Thus, for each $\varepsilon > 0$, there exists $n \; \in \mathbb{N}$ such that for all $m, n \geq n$ we have $d(x_m, x_n) \leq d(x_m, x_n) + T(u(x_m), u(x_n)) = \varrho(x_m, x_n) \quad \varepsilon$. So (x_n) is also a Cauchy sequence in (X, d) and therefore has a unique limit x in (X, d). In particular, we have $x_n \to x$ in (X, d) and also $u(x_n) \to u(x)$ and $T(u(x_n), u(x)) \to T(u(x), u(x)) = u(x)$. We have to show that $\varrho(x_n, x)$ converges to 0 as $n \to$. For all $n \in \mathbb{N}$ we obtain $\varrho(x_n, x) = d(x_n, x) + T(u(x_n), u(x)) \to u(x) = u_\varrho(x)$, and it remains to show that $\varrho(x, x) = 0$. But this follows from the fact that (x_n) is a Cauchy sequence, since this implies that $u(x_n) = u_\varrho(x_n) = \varrho(x_n, x_n) \to 0$ as $n \to$, hence by continuity of u we obtain $u(x) = 0$. ■

We can also obtain a partial converse of Proposition 3.1.3.

3.1.5 Proposition Let (X, ϱ) be a d-metric space which satisfies condition (Piv) from Definition 1.4.12 and let $T : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ be a symmetric operator such that $T(x, x) = x$ for all $x \in \mathbb{R}^+$ and which satisfies the inequality

$$T(x, y) \geq T(x, z)+ \; T(z, y) - T(z, z)$$

for all $x, y, z \in \mathbb{R}^+$. Then (X, d) with

$$d(x, y) = \varrho(x, y) - T\left(u_\varrho(x), u_\varrho(y)\right)$$

is a pseudometric space.

Proof: (Mi) For all $x \in X$ we have $d(x, x) = \varrho(x, x) - u_\varrho(x) = 0$.
(Miii) Obvious by symmetry of ϱ and T.
(Miv) For all $x, y \in X$ we obtain

$$\begin{aligned} d(x, y) &= \varrho(x, y) - T(u_\varrho(x), u_\varrho(y)) \\ &\leq \varrho(x, z)+ \; \varrho(z, y) - \varrho(z, z) - (T(u_\varrho(x), u_\varrho(z)) \quad T(u_\varrho(z), u_\varrho(y)) - u_\varrho(z)) \\ &= \varrho(x, z) - T(u_\varrho(x), u_\varrho(z)) \quad \varrho(z, y) - T(u_\varrho(z), u_\varrho(y)) \\ &= d(x, z)+ \; d(z, y) \end{aligned}$$

■

An example of a natural operator T which satisfies the requirements of Propositions 3.1.3, 3.1.4 and 3.1.5 is

$$T : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+ : (x, y) \to \frac{1}{2}(x \quad y),$$

cf. [Mat92].

We discuss a few more examples of d-metrics which are partly taken from [Mat92].

3.1.6 Example Let d be the metric $d(x, y) = \frac{1}{2}|x - y|$ on $\mathbb{R}^+$, let $u : \mathbb{R}^+ \to \mathbb{R}^+$ be the identity function, and define $T(x, y) = \frac{1}{2}(x \quad y)$. Then ϱ as defined in Proposition 3.1.3 is a d-metric and $\varrho(x, y) = \frac{1}{2}|x - y| \quad \frac{1}{2}(x \quad y) = \max\{x, y\}$ for all $x, y \in \mathbb{R}^+$.

3.1.7 Example Let $\mathcal{I}$ be the set of all closed intervals on $\mathbb{R}$. Then $d : \mathcal{I} \times \mathcal{I} \to \mathbb{R}^+$ defined by

$$d([a, b], [c, d]) = \frac{1}{2}\left(|a - c| \quad |b - d|\right)$$

is a metric on $\mathcal{I}$. Let $u : \mathcal{I} \to \mathbb{R}^+$ be defined by

$$u([a, b]) = b - a$$

and let T be defined as in Example 3.1.6. Then the construction in Proposition 3.1.3 yields a d-metric ϱ such that

$$\varrho([a, b], [c, d]) = \max\{b, d\} - \min\{a, c\}$$

for all $[a,b],[c,d] \in \mathcal{I}$.
Indeed, we obtain

$$\begin{aligned}
\varrho([a,b],[c,d]) &= d([a,b],[c,d]) \quad \frac{1}{2}b - \frac{1}{2}a \quad \frac{1}{2}d - \frac{1}{2}c \\
&= \frac{1}{2}\left(|b-d| \quad b \quad d \quad |a-c| - a - c\right) \\
&= \frac{1}{2}\left(|b-d| \quad b \quad d)\right) \quad \frac{1}{2}\left(|a-c| - (a \quad c)\right) \qquad +(\\
&= \max\{b,d\} - \min\{a,c\}.
\end{aligned}$$

3.1.8 Example $(\mathbb{R}^+, \varrho)$ where $\varrho : (x,y) \to x \quad y$ is a dislocated metric space.

The following proposition gives an alternative way of obtaining d-ultrametrics from ultrametrics. We will apply this in Section 5.2.

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

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

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

Proof: (Mii) and (Miii) are obvious.
(Miv′) We obtain for all $x,y,z \in X$

$$\begin{aligned}
\varrho(x,y) &= \max\{d(x,y), u(x), u(y)\} \\
&\leq \max\{d(x,z), d(z,y), u(x), u(y)\} \\
&\leq \max\{d(x,z), u(x), u(z), d(z,y), u(y)\} \\
&= \max\{\varrho(x,z), \varrho(z,y)\}.
\end{aligned}$$

For completeness, let (x_n) be a Cauchy sequence in (X,ϱ). Then (x_n) is a Cauchy sequence in (X,d) and converges to some $x \in X$. We then obtain $\varrho(x_n,x) = \max\{d(x_n,x), u(x_n), u(x)\} \to u(x)$ as $n \to \quad$. As in the proof of Proposition 3.1.4 we obtain $u(x) = 0$ which completes the proof. ■

We investigate the relationship between the Matthews theorem 1.4.6 and the Banach contraction mapping theorem 1.2.2.

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

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

3.1.11 Proposition Let (X, ϱ) be a d-metric space and define $d : X \times X \to \mathbb{R}$ by $d(x, y) = \varrho(x, y)$ for $x = y$ and $d(x, x) = 0$ for all $x \in X$. If the metric d is complete, so is ϱ, and if f is a contraction relative to ϱ then f is also a contraction relative to d.

Proof: If (x_n) is a Cauchy sequence in ϱ, then for all ε there exists n such that $\varrho(x_k, x_m) \quad \varepsilon$ for all $k, m \geq n$. Consequently, we also obtain $d(x_k, x_m) \quad \varepsilon$ for all $k, m \geq n$, and since d is complete, the sequence (x_n) converges in d to some x and $d(x_n, x) \to 0$ as $n \to$. It remains to show that $\varrho(x_n, x) \to 0$ as $n \to$.

We consider two cases.

(1) Assume that the sequence (x_n) is such that there exists n with $x_m = x$ for all $m \geq n$. Then $\varrho(x_m, x) = d(x_m, x)$ for all $m \geq n$, i.e. $\varrho(x_m, x) \to 0$, and hence $\varrho(x_n, x) \to 0$.

(2) Assume that there exist infinitely many $n_k \in \mathbb{N}$ such that $x_{n_k} = x$. Since (x_n) is a Cauchy sequence with respect to ϱ we obtain $\varrho(x_{n_k}, x) \qquad$ for all $\varepsilon > 0$, i.e. $\varrho(x, x) = 0$. Hence $\varrho(x_n, x) = d(x_n, x)$ for all $n \in \mathbb{N}$ as required.

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

3.1.12 Proposition Let (X, ϱ) be a complete d-metric space and define $d : X \times X \to \mathbb{R}$ by $d(x, y) = \varrho(x, y)$ for $x = y$ and $d(x, x) = 0$ for all $x \in X$. Then the metric d is complete, and if f is a contraction relative to d then f is not necessarily a contraction relative to ϱ.

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

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

We can now prove the Matthews theorem 1.4.6 by using the Banach contraction mapping theorem 1.2.2.

Alternative proof of Theorem 1.4.6 Let (X, ϱ) be a complete d-metric space and f a contraction relative to ϱ. Define d as above. Then d is a complete metric and f is a contraction relative to d. So f has a unique fixed point by the Banach contraction mapping theorem. ■

3.2 Domains as Generalized Ultrametric Spaces

It is our intention here to cast domains into ultrametric spaces. Usually, domains are endowed with the Scott topology, which is one of the T (but not T_1) topologies of interest in theoretical computer science. However, as we will see, domains can be endowed with the structure of a spherically complete ultrametric space. This is not something normally considered in domain theory. However, given that there are many ultrametrics which are useful in theoretical computer science, it suggests that a study of the properties of generalized ultrametric spaces, as carried out e.g. in [Kuh99, Rib96, BMPC99, PC90, PCR93, PCR00c, PCR00b, PCR00a], from this viewpoint is worthy of consideration.

We now cast an arbitrary domain into an ultrametric space. For this purpose, let γ denote an arbitrary countable ordinal, and let Γ_γ denote the set $\{2^{-\alpha} \mid \alpha < \gamma\}$ of symbols $2^{-\alpha}$ ordered by $2^{-\alpha}$ $2^{-\beta}$ if and only if β .

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

Then (D, d_r) is an ultrametric space said to be *induced by* r. The definition of d_r is a variation of a construction made by M.B. Smyth in [Smy91, Example 5], and applied to level mappings in logic programming in [Sed97]. Indeed, the intuition behind d_r is that two elements x and y of the domain D are "close" if they dominate the same compact elements up to a certain rank (and hence *agree* in this sense up to this rank); the higher the rank giving agreement, the closer are x and y. Furthermore, (D, d_r) is spherically complete. The proof of this claim does not make use of the existence of a bottom element of D, so this requirement can be omitted. The main idea of the proof is captured in the following lemma which shows that chains of balls give rise to chains of elements in the domain. It depends on the following elementary facts, see also Lemma 1.3.3.

3.2.2 Fact (1) If $\gamma \leq \delta$ and $x \in B_\delta(y)$, then $B_\gamma(x)$ $B_\delta(y)$. Hence every point of a ball is also its centre.
(2) If $B_\gamma(x) \subset B_\delta(y)$, then $\delta \leq \gamma$ (thus γ , if Γ is totally ordered).

It will simplify notation in the following proof to denote the ball $B_{2^{-\alpha}}(x)$ by $B^\alpha(x)$.

3.2.3 Lemma Let $B^\beta(y)$ and $B^\alpha(x)$ be arbitrary balls in (D, d_r). Then the following statements hold.

(1) For any $z \in B^\beta(y)$, we have $\{c \in \mathrm{approx}(z) \mid r(c) \quad \beta\} = \{c \in \mathrm{approx}(y) \mid r(c) \quad \}$. $<\beta$

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

(3) $B_\beta \in B^\beta(y)$ and $B_\alpha \in B^\alpha(x)$.

(4) Whenever $B^\alpha(x) \quad B^\beta(y)$, we have $B_\beta \sqsubseteq B_\alpha$.

Proof: (1) Since $d_r(z, y) \leq 2^{-\beta}$, the first statement follows immediately from the definition of d_r.
(2) Since the set $\{c \in \mathrm{approx}(z) \mid r(c) \quad \}$ is bounded by z, for any z and β, the second statement follows immediately from the consistent completeness of D.
(3) By definition, we obtain $B_\beta \sqsubseteq y$. Since B_β and y agree on all $c \in D_\mathrm{c}$ with $r(c) \quad$, the first statement in (3) holds, and the second similarly. $<\beta$
(4) First note that $x \in B^\beta(y)$, so that $B^\beta(y) = B^\beta(x)$ and the hypothesis can be written as $B^\alpha(x) \quad B^\beta(x)$. We consider two cases.
(i) If $\beta \leq \alpha$, then using (1) and noting again that $x \in B^\beta(y)$ we get $B_\beta = \sup\{c \in \mathrm{approx}(y) \mid r(c) \quad \beta\} = \sup\{c \in \mathrm{approx}(x) \mid r(c) \quad \beta\} \sqsubseteq \sup\{c \in \mathrm{approx}(x) \mid r(c) \quad \} = B_\alpha$ as required. $<\alpha$
(ii) If $\alpha \quad \beta$, then we cannot have $B^\alpha(x) \subset B^\beta(x)$ and we therefore obtain $B^\alpha(x) = B^\beta(x)$ and consequently $B^\alpha(B_\beta) = B^\beta(B_\beta) = B^\beta(B_\alpha)$ using (3). With the argument of (i) and noting this time that $y \in B^\alpha(x)$, it follows that $B_\alpha \sqsubseteq B_\beta$. We want to show that $B_\alpha = B_\beta$. Assume in fact that $B_\alpha \sqsubset B_\beta$. Since any point of a ball is its centre, we can take $z = B_\beta$ in (2), twice, to obtain $B_\beta = \sup\{c \in \mathrm{approx}(B_\beta) \mid r(c) \quad \beta\}$ and $B_\alpha = \sup\{c \in \mathrm{approx}(B_\beta) \mid r(c) \quad \alpha\}$. Thus, the supposition $B_\alpha \sqsubset B_\beta$ means that $\sup\{c \in \mathrm{approx}(B_\beta) \mid r(c) \quad \alpha\} \sqsubset \sup\{c \in \mathrm{approx}(B_\beta) \mid r(c) \quad \beta\}$. Since $\{c \in \mathrm{approx}(B_\beta) \mid r(c) \quad \alpha\} \quad \{c \in \mathrm{approx}(B_\beta) \mid r(c) \quad \beta\}$, there must be some $d \in \{c \in \mathrm{approx}(B_\beta) \mid r(c) \quad \beta\}$ with $d \not\sqsubseteq \sup\{c \in \mathrm{approx}(B_\beta) \mid r(c) \quad \alpha\} = B_\alpha$. Thus, there is an element $d \in D_\mathrm{c}$ with $r(d) \quad \beta$ satisfying $d \not\sqsubseteq B_\alpha$ and $d \sqsubseteq B_\beta$. This contradicts the fact that $d_r(B_\alpha, B_\beta) \leq 2^{-\beta}$. Hence, $B_\alpha \not\sqsubset B_\beta$, and since $B_\alpha \sqsubseteq B_\beta$, it follows that $B_\alpha = B_\beta$ and therefore that $B_\beta \sqsubseteq B_\alpha$ as required. ■

3.2.4 Theorem The ultrametric space (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 $B^\alpha(x_\alpha)$ be an arbitrary ball in the chain. It suffices to show that $B \in B^\alpha(x_\alpha)$. Since $B_\alpha \in B^\alpha(x_\alpha)$, we have $d_r(B_\alpha, x_\alpha) \leq 2^{-\alpha}$. But d_r is a generalized ultrametric and so it suffices to show that $d_r(B, B_\alpha) \leq 2^{-\alpha}$. For every compact element $c \sqsubseteq B_\alpha$, we have $c \sqsubseteq B$ by construction of B. Now let $c \sqsubseteq B$ with $c \in D_\mathrm{c}$ and $r(c) \quad \alpha$. We have to show that $c \sqsubseteq B_\alpha$. Since c is compact and $c \sqsubseteq B$, there exists B_β in the chain with $c \sqsubseteq B_\beta$. If $B^\alpha(x_\alpha) \quad B^\beta(x_\beta)$, then $B_\beta \sqsubseteq B_\alpha$ by Lemma 3.2.3 and therefore $c \sqsubseteq B_\alpha$. If $B^\beta(x_\beta) \subset B^\alpha(x_\alpha)$, then $\alpha \quad \beta$, and since $c \sqsubseteq B_\beta$, c is an element of the set $\{c \in \mathrm{approx}(x_\beta) \mid r(c) \quad\quad \} = \{c \in \mathrm{approx}(x_\alpha) \mid r(c) \quad \alpha\}$. Since B_α is the supremum of the latter set, we have $c \sqsubseteq B_\alpha$ as required. ■

This result will be applied in Section 5.1.

3.3 Generalized Ultrametric Spaces as Domains

We will give an alternative proof of the Prieß-Crampe and Ribenboim theorem which is inspired by [EH98], where the Banach contraction mapping theorem 1.2.2 was proven from the Kleene theorem 1.1.3. We will prove the Prieß-Crampe and Ribenboim theorem using the Knaster-Tarski theorem 1.1.7. For this purpose, we will again impose the condition on the generalized ultrametric space (X, d, Γ), that Γ is of the form $\{2^{-\alpha} \mid \alpha \leq \gamma\}$ for some ordinal γ, ordered as in Section 3.2 and in Definition 2.3.5 by $2^{-\alpha} \leq 2^{-\beta}$ if $\beta \leq \alpha$. Such a generalized ultrametric space will henceforth be called a gum with *ordinal distances*. Recall that we denote $2^{-\gamma}$ by 0.

The main technical tool which was employed in [EH98] is the space of formal balls associated with a given metric space. We will extend this notion to generalized ultrametrics.

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

3.3.1 Proposition The set $\mathcal{B}X$ is partially ordered by $\sqsubseteq$. Moreover, X is spherically complete if and only if $\mathcal{B}X$ is chain-complete.

Proof: Let X be spherically complete and let $[(x_\beta, \beta)]$ be an ascending chain in $\mathcal{B}X$. Then $B_\beta(x_\beta)$ is a chain of balls in X with non-empty intersection, and let $x \in \quad B_\beta(x_\beta)$. Then $d(x_\beta, x) \leq \beta$ for all β. Hence the chain $[(x_\beta, \beta)]$ in $\mathcal{B}X$ has $[(x, 0)]$ as an upper bound. Now consider the set A of all $\alpha \in \Gamma$ such that $[(x, \alpha)]$ is an upper bound of $[(x_\beta, \beta)]$. Since we are working with ordinal distances only, the set A has a supremum γ, and hence $[(x, \gamma)]$ is a least upper bound of the chain $[(x_\beta, \beta)]$.

Now let $\mathcal{B}X$ be chain-complete and let $(B_\beta(x_\beta))_{\beta\in\Lambda}$, where $\Lambda \quad \Gamma$, be a chain of balls in X. Then $[(x_\beta, \beta)]$ is an ascending chain in $\mathcal{B}X$ and has a least upper bound (x, γ), and hence $B_\gamma(x) \qquad B_\beta(x_\beta)$. ■

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

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

Given a strictly contracting mapping f on a generalized ultrametric space (X, d, Γ) with ordinal distances, we define a function $\mathcal{B}f : \mathcal{B}X \to \mathcal{B}X$ by

$$\left(x, 2^{-\alpha}\right) \to \begin{array}{ll} \left(f(x), 2^{-(\alpha+1)}\right) & \text{if } 2^{-\alpha} = 0 \\ (f(x), 0) & \text{if } 2^{-\alpha} = 0. \end{array}$$

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

Proof: Let $(x, 2^{-\alpha}) \sqsubseteq (y, 2^{-\beta})$, so that $d(x, y) \le 2^{-\alpha}$ and $\alpha \le \beta$. If $2^{-\alpha} = 0$, there is nothing to show, so assume $2^{-\alpha} = 0$. It only remains to show that $d(f(x), f(y)) \le 2^{-(\alpha+1)}$, which holds since f is strictly contracting, and that $\alpha \quad \le \beta \quad$ if $2^{-\beta} = 0$, and that $\alpha \quad \le \beta$ if $2^{-\beta} = 0$ and $\alpha = \beta + 1$, which are easy to see. ■

Alternative proof of Theorem 1.3.4 Let (X, d, Γ) be a spherically complete generalized ultrametric space with ordinal distances, and let $f : X \to X$ be strictly contracting. Then $\mathcal{B}X$ is a chain-complete partially ordered set, and $\mathcal{B}f$ is a monotonic mapping on $\mathcal{B}X$. For $B \in \mathcal{B}X$, we denote by $\uparrow B$ the upper cone of B , that is, the set of all $B \in \mathcal{B}X$ with $B \sqsubseteq B$.

Let $x \in X$ be arbitrarily chosen, assume without loss of generality that $x = f(x)$, and let α be an ordinal such that $d(x, f(x)) = 2^{-\alpha}$. Then $(x, 2^{-\alpha}) \sqsubseteq \left(f(x), 2^{-(\alpha+1)}\right)$, and by monotonicity of $\mathcal{B}f$ we obtain that $\mathcal{B}f$ maps $\uparrow[(x, 2^{-\alpha})]$ into itself. Since $\uparrow[(x, 2^{-\alpha})]$ is a chain-complete partial order with bottom element $[(x, 2^{-\alpha})]$, we obtain by the Knaster-Tarski theorem 1.1.7 that $\mathcal{B}f$ has a least fixed point in $\uparrow[(x, 2^{-\alpha})]$ which we will denote by B .

It is clear by definition of $\mathcal{B}f$ that B must be maximal in $\mathcal{B}X$, and hence is of the form $[(x\ , 0)]$. From $\mathcal{B}f[(x\ , 0)] = [(x\ , 0)]$ we obtain $f(x\ ) = x$, so that x is a fixed point of f.

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

3.4 Generalized Ultrametrics and Dislocated Generalized Ultrametrics

We investigate the relationship between the Prieß-Crampe and Ribenboim theorem 1.3.4 and it's dislocated version, Theorem 1.5.2.

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

Proof: The proof is straightforward following Proposition 3.1.10. ■

3.4.2 Proposition Let (X, ϱ) be a dislocated generalized ultrametric space and define $d : X \times X \to \mathbb{R}$ by $d(x, y) = \varrho(x, y)$ for $x = y$ and $d(x, x) = 0$ for all $x \in X$. If d is spherically complete then ϱ is spherically complete, and if f is strictly contracting relative to ϱ then f is also strictly contracting relative to d.

Proof: We first show that non-empty balls in ϱ contain all their midpoints. So let $\{y \mid \varrho(x, y) \leq \alpha\}$ be some non-empty ball in ϱ with midpoint x. Then there exists some $z \in \{y \mid \varrho(x, y) \leq \alpha\}$ and we obtain $\varrho(x, x) \leq \varrho(x, z)$ by (Uiv) and since $\varrho(x, z) \leq \alpha$ we have $x \in \{y \mid \varrho(x, y) \leq \alpha\}$. Hence, every non-empty ball in ϱ is also a ball with respect to d.

Now let $\mathcal{B}$ be a chain of non-empty balls in ϱ. Then $\mathcal{B}$ is also a chain of balls in d and has non-empty intersection by spherical completeness of d as required.

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

3.4.3 Proposition Let (X, ϱ) be a spherically complete dislocated generalized ultrametric space and define $d : X \times X \to \mathbb{R}$ by $d(x, y) = \varrho(x, y)$ for $x = y$ and $d(x, x) = 0$ for all $x \in X$. Then d is spherically complete, and if f is strictly contracting relative to d then f is not necessarily strictly contracting relative to ϱ.

Proof: Let $\mathcal{B}$ be a chain of balls in d. If $\mathcal{B}$ contains a ball $B = \{x\}$ for some $x \in X$, then x is in the intersection of the chain. So assume that all balls in $\mathcal{B}$ contain more than one point.

Now let $B_\gamma(x_m) = \{x \mid d(x, x_m) \leq \gamma\}$ be a ball in $\mathcal{B}$ and let $x_m = z \in B_\gamma(x_m)$. Then $\varrho(x_m, x_m) \leq \varrho(x_m, z) = d(x_m, z) \leq \alpha$, hence $B_\gamma(x_m) = \{x \mid \varrho(x, x_m) \leq \gamma\}$. It follows that $\mathcal{B}$ is also a chain of balls in ϱ and has non-empty intersection as required.

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

We can now use Theorem 1.3.4 to give an easy proof of Theorem 1.5.2.

Alternative proof of Theorem 1.5.2 Using Proposition 3.4.1, we obtain a generalized ultrametric space (X, d), which is spherically complete by Proposition 3.4.3. By Proposition 3.4.2, the function f is strictly contracting relative to d. Hence, by Theorem 1.3.4, f has a unique fixed point. ■

We close by giving two constructions of d-gums from gums.

3.4.4 Proposition Let (X, d, Γ) be a generalized ultrametric space with ordinal distances and let $u : X \to \Gamma$ be a function. Then the distance function

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

is a dislocated generalized ultrametric on X.

Proof: (Ui) and (Uiii) are trivial. For (Uiv) see the proof of Proposition 3.1.9. ■

This result will be applied in Section 5.4.

3.4.5 Proposition Let (X, d, Γ) be a generalized ultrametric space with ordinal distances, let $z \in X$, and define a function

$$\varrho : X \times X \to \Gamma : (x, y) \to \max\{d(x, z), d(y, z)\}.$$

Then (X, ϱ, Γ) is a dislocated generalized ultrametric space. Furthermore, if (X, d) is spherically complete, then so is (X, ϱ).

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

This result will be applied in Section 5.5.

3.5 Summary and Further Work

We have covered two main themes in this chapter, which are (1) the relationships between the dislocated and non-dislocated versions of the Banach contraction mapping theorem 1.2.2 and the Prieß-Cramps and Ribenboim theorem 1.3.4, resulting in alternative proofs of the Matthews theorem 1.4.6 and Theorem 1.5.1, covered in Sections 3.1 and 3.4 and (2) relationships between Scott-Ershov domains and generalized ultrametric spaces, covered in Sections 3.2 and 3.3.

The proof of the Matthews theorem 1.4.6 in Section 3.1 involved the casting of a d-metric into a metric, hence implicitly allows to introduce a metrizable topology on the d-metric space. In Section 1.4, in the paragraph after Definition 1.4.12, we noted that partial and weak partial metrics, which are also d-metrics, allow for a natural topology obtained from open balls. Thus we have two natural topologies on partial and weak partial metrics, and an obvious question is how these two relate. Further investigations on (weak) partial metric spaces are currently being undertaken by different authors, e.g. in [EH98, Wac00], and domain-theoretic arguments naturally come into view in this context.

Generalized ultrametrics have, to the best of our knowledge, not been studied in the context of domain theory beforehand. Sections 3.2 and 3.3 provide a first step towards such investigations. The domain-theoretic proof of the Prieß-Crampe and Ribenboim theorem 1.3.4 in Section 3.3, for example, suggests the possibility of a domain-theoretic treatment of non-monotonic operators in logic programming, possibly related to the work of [RZ98, ZR97a, ZR97b, ZR98], where the operator corresponding to the stable model semantics [GL88], cf. Chapter 7, is studied from a domain-theoretic point of view. In the publications just mentioned, operators in three-valued logic as in [Fit85] play an important role, and they will also be considered in this thesis in Chapter 6.

We finally note that the constructions used for casting domains into generalized ultrametrics as in Section 3.2, and for casting generalized ultrametrics into chain-complete partial orders as in Section 3.3, are not inverse to each other, and it remains to be investigated under what conditions inverses can be found.

Part II

Logic Programming Semantics

Chapter 4

Topologies for Logic Programming Semantics

If P is a definite logic program, then the operator T_P is continuous in the Scott topology on I_P, and has a least fixed point due to the Kleene theorem 1.1.3. This fixed point corresponds very well to the procedural semantics of the program under logic programming systems like Prolog [Llo88]. In the case of normal programs, the single-step operator is no longer monotonic, and the Scott topology is insufficent for analyzing its behaviour. An alternative to the Scott topology in this case is the Cantor topology on I_P, also called the atomic topology Q. The results presented in this part of the thesis support the claim that Q is the major alternative choice of a topology for logic programming semantics.

In Section 4.1, we will shortly review the Scott topology on I_P in the form in which it was presented in [Sed95]. In Section 4.2, we discuss the atomic topology and present some first results which support the claim that it is a highly suitable topology for our analysis. In Section 4.3, we will introduce a generalization of the atomic topology for multi-valued logics.

In this chapter, we will work under fixed but arbitrary preinterpretations.

4.1 Scott Topology (Positive Atomic Topology)

We shortly review the Scott topology on the space of all interpretations of a program. For proofs of the results in this section, see [Sed95].

4.1.1 Definition 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 a subbase of a topology, the *positive atomic topology* Q^+ on I_P.

Note that a basic open set in Q^+ is of the form $\mathcal{G}(A_1) \cap \quad \cap \mathcal{G}(A_n)$, which we will write as $\mathcal{G}(A_1, \dots, A_n)$. If B_P is countable, e.g. in the case when the preinterpretation is Herbrand, we note that Q^+ is second countable.

The topology Q^+ can be characterized by convergence using the following proposition.

4.1.2 Proposition A net (I_λ) converges in Q^+ to $I \in I_P$ if and only if every element of I is eventually an element of I_λ, i.e. if and only if for each $A \in B_P$ there exists λ such that $A \in I_\lambda$ for all $\lambda \geq \lambda$.

4.1.3 Proposition The positive atomic topology Q^+ on I_P coincides with the Scott topology on I_P.

4.1.4 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 $\mathrm{gl}(I_n) = \bigcup_{k \geq k_0} I_k$.

If P is a definite program, then the operator T_P is Scott-continuous on I_P, hence admits a least fixed point M_P by the Kleene theorem 1.1.3. The supported model M_P is also the least model of P and is interpreted as the intended meaning of P, since it corresponds very well to the procedural behaviour under logic programming systems [Llo88].

In the special case of Herbrand preinterpretations, the positive atomic topology is called the *positive query topology*, which was introduced and analyzed in [Bat89, BS89b, BS89a], and only later on generalized to arbitrary preinterpretations.

4.2 Cantor Topology (Atomic Topology)

We introduce the atomic topology due to [Sed95] and prove some first results which support the claim that it is a very suitable topology for the analysis of non-monotonic semantic operators.

4.2.1 Definition Let P be a logic program. The set $\{\mathcal{G}(A) \mid A \in B_P\}$ $\{\mathcal{G}(\neg A) \mid A \in B_P\}$, where $\mathcal{G}(A) = \{I \in I_P \mid A \in I\}$ and $\mathcal{G}(\neg A) = \{I \in I_P \mid A \in I\}$, is a subbase of a topology, the *atomic topology* Q on I_P.

The atomic topology was first developed, analyzed, and applied in the special case of Herbrand preinterpretations in [Bat89, BS89b, BS89a], where it was called the *query topology*, and later on generalized to arbitrary preinterpretations in [Sed95].

Note that the basic open sets of Q are of the form $\mathcal{G}(A_1) \cap \cdot \quad \cdot \cap \mathcal{G}(A_k) \cap \mathcal{G}(\neg B_1) \cap$ $\cap\, \mathcal{G}(\neg B_l)$, which we will write as $\mathcal{G}(A_1, \ldots, A_k, \neg B_1, \ldots, \neg B_l)$. Clearly, Q is finer than Q^+ and is second countable if the domain of the preinterpretation is countable.

The atomic topology can be characterized by convergence using the following result due to [Sed95].

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

We recall two further results on the atomic topology due to [Sed95].

4.2.3 Proposition 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.

4.2.4 Theorem (I_P, Q) is a totally disconnected compact Hausdorff space. It is also second countable and metrizable if the domain of the chosen preinterpretation is countable. It is homeomorphic to the Cantor set in the real line, if B_P is countably infinite.

We will now present some results which underline the importance of the atomic topology as an alternative to the Scott topology in a non-monotonic context.

4.2.5 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 $T_P(M) \quad M$. Let $A \in T_P(M)$. By definition of T_P, there exists a ground instance $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ of a clause in P with $A_k \in M$ and $B_l \in M$ for $k = 1, \ldots, k_1$, $l = 1, \ldots, l_1$. By Proposition 4.2.2, there is an $n \in \mathbb{N}$, such that for all $n \geq n$, $A_k \in T_P^n(I)$ and $B_l \in T_P^n(I)$ for all k, l. By definition of T_P and the above clause we have that $A \in T_P^m(I)$ for all $m \geq n$. Hence, $A \in T_P^n(I)$ eventually and therefore, by Proposition 4.2.2 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. ∎

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 4.2.5, 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], and we borrow the following result, which will be of use in Chapter 9.

4.2.6 Theorem Let P be a normal logic program. Then T_P is continous in Q if and only if, for each $I \in I_P$ and for each $A \in B_P$ with $A \in T_P(I)$, either there is no clause in P with head A or there is a finite set $S(I, A) = \{A_1, \ldots, A_k, B_1, \ldots, B_k\}$ of elements of B_P with the following properties:

(i) $A_1, \ldots, A_k \in I$ and $B_1, \ldots, B_k \in I$.

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

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

4.2.7 Theorem Let P be a normal logic program and let $I \in I_P$ be such that the sequence (I_n), with $I_n = T_P^n(I)$, 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 $M \quad T_P(M)$. So let $A \in M$. By convergence in Q and Proposition 4.2.2, there exists $n \in \mathbb{N}$ such that $A \in T_P^n(I)$ for all $n \geq n$. By hypothesis, there are only finitely many clauses in ground(P) with head A. Let C be the (finite) set of all atoms occurring in positive body literals and D the (finite) set of all atoms occurring in negative body literals of those clauses. Let $C_1 = C \cap M$ and $D_1 = D \setminus M$. Since $I_n \to M$ in Q, there is an $n_1 \in \mathbb{N}$ such that $C_1 \quad I_n$ and $D_1 \quad B_P \setminus I_n$ for all $n \geq n_1$. Since $A \in T_P\left(I_{\max\{n_0, n_1\}}\right)$, there is a clause $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ in ground(P) with $A_k \in C_1 \quad M$ and $B_l \in D_1 \quad B_P \setminus M$ for $k = 1, \ldots, k_1$, $l = 1, \ldots, l_1$. Hence $A \in T_P(M)$ as required. ∎

In the sequel, it will often be necessary to tranfinitely iterate the operator T_P before a fixed point is reached. The following result is an obvious, but fundamental generalization of Theorem 4.2.5.

4.2.8 Theorem Let P be a normal logic program and let $I \in I_P$ and define, for each limit ordinal α,

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

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

Proof: The proof is a straightforward adaptation of the proof of Theorem 4.2.5 and is omitted. ∎

4.3 Generalized Atomic Topologies

We generalize the atomic topology to multivalued logics.

In the following, let P be a normal logic program. We consider logics, underlying P, with finitely many truth values $t\ , t_1, \ldots, t_{n-1}$. An *interpretation* under such a logic is a tuple $I = (I\ , \ldots, I_{n-1})$ where each I_i is a set of ground atoms from P such that all I_i are mutually disjoint and $\quad I_i = B_P$, where B_P is the set of all ground atoms from the first order language underlying P. For every $i = 0, \ldots, n-1$, each atom in I_i has truth value t_i under I, and we write $v_I(A) = t_i$ for $A \in I_i$. The truth value $t\ $ will be abbreviated as t and we say that an atom A with $v_I(A) = t$ is *true* in I. The function $v : I_P \times B_P : (I, A) \to v_I(A)$, expanded to formulas as second arguments using suitable truth tables for the logical connectives, is called the *valuation function* of the logic. The set of all interpretations of P will be denoted by $I_{P,n}$.

4.3.1 Definition An interpretation $I \in I_{P,n}$ is called a *model* of P if $v_I(C) = t$ for every ground instance C of any clause in P.

We define a topology on $I_{P,n}$ as follows.

4.3.2 Definition Identify $I_{P,n}$ with the set $\{v_I : I \in I_{P,n}\}$. There obviously is a bijective correspondence between the two sets by each I corresponding to v_I. Endowing $\{t\ , \ldots, t_{n-1}\}$ with the discrete topology, we obtain a product topology $\mathcal{Q}$ on $I_{P,n}$ which will be called the *generalized atomic topology*.

Topological Properties

The following two propositions follow from well-known results from elementary topology [Wil70]. Note that $\mathcal{Q}$ is a topology of pointwise convergence since it is a product topology of the discrete topology on a finite set.

4.3.3 Proposition For $A \in B_P$ and t_i a truth value, let $\mathcal{G}(A, t_i) = \{I \in I_{P,n} \mid v_I(A) = t_i\}$. Then $\mathcal{Q}$ is the topology generated by the subbase $\{\mathcal{G}(A, t_i) \mid A \in B_P, i \in \{0, \ldots, n-1\}\}$.

4.3.4 Proposition A net I_λ in $I_{P,n}$ converges in $\mathcal{Q}$ if and only if for every $A \in B_P$ there exists some λ_A such that $v_{I_\lambda}(A)$ is constant for all $\lambda \geq \lambda_A$. In this case, the limit I of the net I_λ is given by $v_I(A) = v_{I_{\lambda_A}}(A)$ for each $A \in B_P$.

We immediately obtain that $\mathcal{Q}$ is indeed a generalization of Q.

4.3.5 Proposition If the chosen logic is the classical (two-valued) logic, then $\mathcal{Q}$ coincides with the atomic topology Q on $I_{P,n} = I_{P,2} = I_P$.

The following theorem also follows from the fact that $\mathcal{Q}$ is a product topology of the discrete topology on a finite set.

4.3.6 Theorem The generalized atomic topology $\mathcal{Q}$ is a totally disconnected compact Hausdorff topology. It is second countable if the domain of the chosen preinterpretation is countable.

Consequence Operators

4.3.7 Definition An operator T on $I_{P,n}$ is called a *consequence operator* for P if for every $I \in I_{P,n}$ the following condition holds: For every ground clause $A \leftarrow \texttt{body}$ in P, where $v_{T(I)}(A) = t_i$, say, and $v_I(\texttt{body}) = t_j$, say, we have that the truth table for $t_i \leftarrow t_j$ yields the truth value true.

Obviously, the single-step operator T_P for normal logic programs P is a consequence operator.

4.3.8 Theorem Let T be a consequence operator for P and let $I \in I_{P,n}$. If $T^m(I)$ converges in $\mathcal{Q}$ to some $M \in I_{P,n}$, then M is a model of P. If, furthermore, T is continuous in $\mathcal{Q}$, then M is a fixed point of T.

Proof: Let $I_m = T^m(I)$ for each m and let $A \in B_P$ with $v_M(A) = t_i$. Then we obtain $v_{I_{k_1}}(A) = t_i$ for all $k_1 \geq k$ for some $k \in \mathbb{N}$ by convergence in $\mathcal{Q}$. Let $A \leftarrow \texttt{body}$ be a ground clause in P. Since T is a consequence operator, we obtain that for any $k_2 > k$, $v_{I_{k_2}}(\texttt{body})$ must have some value t_j such that $t_i \leftarrow t_j$ yields truth value true. Since $\texttt{body}$ is a finite conjunction of ground atoms, and since I_m converges in $\mathcal{Q}$, there must therefore exist some $l \in \mathbb{N}$, chosen large enough, such that for *all* $l \geq l$, $v_{I_l}(\texttt{body})$ evaluates to some t_j which is independent of l and such that $t_i \leftarrow t_j$ yields truth value true. Consequently, again by convergence in $\mathcal{Q}$, the clause $A \leftarrow \texttt{body}$ evaluates to true under M. Since the clause was arbitrarily chosen, M is a model of P.

If T is continuous in $\mathcal{Q}$, we obtain $M = \lim T^{n+1}(I) = T(\lim T^n(I)) = T(M)$. ■

4.3.9 Corollary Let T be a consequence operator, P be a normal logic program, and M be a fixed point of T. Then M is a model of P.

Proof: Since the sequence $T^n(M)$ is constant, it follows by Theorem 4.3.8 that M is a model of P. ■

Continuity

4.3.10 Definition Let $A \in B_P$ and denote by $\mathcal{B}_A$ the set of all body atoms of clauses with head A that occur in ground(P). A consequence operator T is called *local* if for every $A \in B_P$ and any two interpretations $I, K \in I_{P,n}$ which agree on all atoms in $\mathcal{B}_A$, we have $v_{T(I)}(A) = v_{T(K)}(A)$.

The restriction of being local imposed on a consequence operator is very weak and is obviously satsified by the single-step operator in classical two-valued logic.

The following definition, which gives a condition which is weaker than the absence of local variables, can be found in [Sed95, Definition 2].

4.3.11 Definition Let C be a clause in P and $A \in B_P$ such that A unifies with the head of C. The clause C is said to be of *finite type relative to* A if C has only finitely many different ground instances with head A. The program P will be said to be of *finite type relative to* A if each clause in P is of finite type relative to A, i.e. if the set of all clauses in ground(P) with head A is finite. Finally, P will be said to be of *finite type* if P is of finite type relative to A for every $A \in B_P$.

4.3.12 Proposition Let P be a normal logic program of finite type and let T be a local consequence operator for P. Then T is continuous in $\mathcal{Q}$.

Proof: Let $I \in I_{P,n}$ be an interpretation and let $G_2 = \mathcal{G}(A, t_i)$ be a subbasic neighbourhood of $T(I)$ in $\mathcal{Q}$, and note that G_2 is the set of all $\in I_{P,n}$ such that $v_K(A) = t_i$. We need to find a neighbourhood G_1 of I such that $T(G_1)$ G_2.

Since P is of finite type, the set $\mathcal{B}_A$ is finite. Hence the set $G_1 = {}_{B \in \mathcal{B}_A} \mathcal{G}(B, v_I(B))$ is a finite intersection of open sets and therefore open. Since each $\in G_1$ agrees with I on $\mathcal{B}_A$, we obtain $v_{T(K)}(A) = v_{T(I)}(A) = t_i$ for each $\in G_1$ by locality of T. Hence, $T(G_1)$ G_2. ■

4.4 Summary and Further Work

We have described different topologies on the space of all interpretations of a logic program: the Scott topology, the atomic topology, and generalized atomic topologies. From this point of view this space, together with some semantic operator associated with a given program, can be interpreted as a topological dynamical system, in a naive sense, and allows us to study these operators in a topological context instead of an order-theoretic one as in the classical case. Such a point of view will be put to work e.g. in Chapter 9, where we will establish some connections between logic programming and artificial neural networks.

The atomic topology provides a very natural notion of convergence on the space of all interpretations, and in fact it is difficult to imagine a reasonable notion of convergence in this context which is not closely related to the characterization in Proposition 4.2.2. As we will see in Chapter 5, if a net converges with respect to any of the generalized metrics studied in this thesis, then this net also converges with respect to Q, although not vice-versa in general. So all the topologies which capture the convergence notions associated with these generalized metrics will be topologies which are finer than the atomic topology.

The generalized atomic topology of Section 4.3 will not be put to much use in the sequel. The general observations made, however, open up the possibility of studying non-monotonic semantic operators on many-valued logics, which is something which has, to our knowledge, not be done before, as semantic operators on many-valued logics are usually designed to be monotonic, as in [Myc84, Fit85, PP90, GRS91, And97, BFMS98, Nai98, CS00]. As a first step towards such investigations, it should be useful to study these monotonic operators in the context of generalized atomic topologies.

Chapter 5

Supported Model Semantics

In this chapter, we will show that some of the fixed-point theorems from Chapter 1 are applicable to the single-step operator T_P under some conditions on the programs P. In particular, we will apply the Prieß-Crampe and Ribenboim theorem 1.3.4 (Section 5.1), the Matthews theorem 1.4.6 (Section 5.3) and Theorem 1.5.1 (Sections 5.4 and 5.5). Since all these fixed-point theorems yield, if applicable, the existence of a unique fixed point for T_P, the conditions which will be imposed on the programs in order to apply the theorems will always have the effect that the programs under consideration have unique supported models, i.e. are *uniquely determined* [BS89b]. Such classes of programs for which all programs in the class have a unique supported model, will be called *unique supported model classes*, and examples are the acyclic programs [Cav89, Bez89, AB90], the locally hierarchical programs [Cav89, Cav91], and the acceptable programs [AP93, AP94, Mar95]. The latter class is important since it has a strong relationship to termination properties under SLDNF-resolution [AP93] and under Chan's constructive negation [Mar96], and we will devote Section 5.2 to a more thorough study of these programs.

We begin with defining the classes of programs which will be studied in this chapter. We will work over arbitrary preinterpretations.

5.0.1 Definition A normal logic program P is called *locally hierarchical* if there exists a level mapping $l : B_P \to \alpha$, for some ordinal α, such that for each clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) and for all $i = 1, \ldots, n$ we have $l(A) > l(L_i)$. If l can be chosen as an ω-level mapping, then P is called *acyclic*.

We note that Program 0.2.1 is acyclic.

The conditions of being locally hierarchical or acyclic are purely syntactical. In [AP93], these conditions have been relaxed to semi-syntactic requirements by employing interpretations with certain conditions. Our remaining definitions follow these lines, and the following one is taken directly from [AP93].

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

1. p *refers to* q if there is a clause in P with p in its head and q in its body.

2. p *depends on* q if (p, q) is in the reflexive, transitive closure of the relation *refers to*.

3. Neg_P denotes the set of predicate symbols in P which occur in a negative literal in the body of a clause in P.

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

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

Let P be a normal logic program, let $l : B_P \to \omega$ be a level mapping and let I be a model of P whose restriction to the predicate symbols in Neg_P^* is a supported model of P^-. Then P is called *acceptable* (with respect to l and I) provided that the following condition holds.

$$\begin{gathered}\text{For each ground instance } A \leftarrow L_1, \dots, L_n \text{ of a clause in } P \\ \text{and for all } i \in \{1, \dots, n\} \text{ we have:} \\ \text{if} \qquad I \models \bigwedge_{j=1}^{i-1} L_j, \qquad \text{then} \qquad l(A) > l(L_i).\end{gathered} \tag{5.1}$$

We recall the following example program from [AP93].

5.0.3 Program Suppose that $\mathcal{G}$ is an acyclic finite graph. Then the program

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

is acceptable but not acyclic. Again, uppercase letters denote variable symbols, while lowercase letters denote constant symbols.

We can further relax Definition 5.0.2 as follows.

5.0.4 Definition A normal logic program P is called Φ^*-*accessible* if and only if there exists a level mapping l for P and a model I for P whose restriction to the predicate symbols in Neg_P^* is a supported model of P^-, such that the following condition holds. For each clause $A \leftarrow L_1, \dots, L_n$ in ground(P), we either have $I \models L_1 \qquad L_n$ and $l(A) > l(L_i)$ for all $i = 1, \dots, n$ or there exists $i \in \{1, \dots, n\}$ such that $I \models L_i$ and $l(A) > l(L_i)$.

We call P Φ^*-*accessible* if it is Φ^*-accessible and l is an ω-level mapping.

P is called Φ-*accessible* if and only if there exists a level mapping l for P and a model I for P such that the following condition holds. Each $A \in B_P$ satisfies either (i) or (ii):

(i) There exists a clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) with head A such that $I \models L_1$ L_n and $l(A) > l(L_i)$ for all $i = 1, \ldots, n$.

(ii) For each clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) with head A there exists $i \in \{1, \ldots, n\}$ such that $I \models L_i$, $I \models A$ and $l(A) > l(L_i)$.

We call P Φ *-accessible* if it is Φ-accessible and l is an ω-level mapping.

5.1 Acyclic Programs and Locally Hierarchical Programs

In this section, we will apply the Banach contraction mapping theorem 1.2.2 to acyclic programs and the Prieß-Crampe and Ribenboim theorem 1.3.4 to locally hierarchical programs. We will also show that the class of all locally hierarchical programs, although syntactically very restricted, is computationally adequate in the sense that each partial recursive function can be computed, under SLDNF-resolution, by such a program, if the use of safe cuts is allowed.

We begin our study of locally hierarchical programs by showing how such a program P can be endowed with a canonical level mapping l_P which is smallest in a certain obvious sense.

5.1.1 Construction Let P be a program which is locally hierarchical with respect to a level mapping l. We define a level mapping l_P on B_P as follows. For every $A \in B_P$ which does not occur as a head in ground(P), let $l_P(A) = 0$. For every $A \in B_P$ which occurs as the head of a unit clause but not as the head of any non-unit clause, let $l_P(A) = 0$. Now let $A \in B_P$ be such that A is the head of some non-unit clause(s) in ground(P). Let $\mathcal{B}_A$ be the collection of body-literals occurring in these clauses. Note that $\mathcal{B}_A$ is finite for every A if P has no local variables. Now suppose that for every $B \in \mathcal{B}_A$, $l_P(B)$ is already defined. Let $M_A = \sup_{B \in \mathcal{B}_A} l_P(B)$ and set $l_P(A) = M_A$, if M_A is a successor ordinal, and set $l_P(A) = M_A$, if M_A is a limit ordinal. Then l_P is obtained by transfinitely iterating this procedure. We will refer to l_P, as defined above, as the *canonical lh-level mapping* of P and, further, γ_P will denote the smallest ordinal α such that $l_P(A) \in \alpha$ for all $A \in B_P$.

5.1.2 Proposition Let P be a program which is locally hierarchical with respect to some level mapping l. Then l_P, as defined above, is a total function on B_P and P is locally hierarchical with respect to l_P. Moreover, if P has no local variables, then $\gamma_P \leq \omega$ and hence P is acyclic.

Proof: First we show that $\mathrm{dom}(l_P) = B_P$. Suppose there is $A \in B_P \setminus \mathrm{dom}(l_P)$. Without loss of generality we can further suppose that $l(A)$ is minimal for A with this property. Then there must be some $B \in \mathcal{B}_A$ with $B \in \mathrm{dom}(l_P)$, otherwise $l_P(A)$ is defined in the process given in Construction 5.1.1. Since P is locally hierarchical, we have $l(B) \quad l(A)$ which contradicts the choice of A with $l(A)$ minimal. Therefore, l_P is a (total) level mapping, and obviously P is locally hierarchical with respect to it. Finally, if P has no local variables, then the set $\mathcal{B}_A$ is finite for every $A \in B_P$, and so l_P maps into ω. Hence, $\gamma_P \leq \omega$. ■

The construction above of the level mapping l_P can be used to determine whether or not a given program P is locally hierarchical, and the following corollary is immediate.

5.1.3 Corollary Let P be an arbitrary normal logic program. Then P is locally hierarchical if and only if $\mathrm{dom}(l_P) = B_P$, where l_P is constructed as in Construction 5.1.1. Furthermore, if P is locally hierarchical, it is locally hierarchical with respect to l_P.

5.1.4 Proposition Let P be a program which is locally hierarchical with respect to a level mapping l. Then for every $A \in B_P$, we have $l_P(A) \leq l(A)$.

Proof: Suppose the conclusion is false. Thus, there is $A \in B_P$ with $l(A) \quad {}_P(A)$, and such that $l(A)$ is minimal. Then, for all $B \in \mathcal{B}_A$, we have $l(B) \quad (A)$ because P is locally hierarchical. Therefore, by minimality of $l(A)$, we have $l(B) \geq l_P(B)$ for all $B \in \mathcal{B}_A$. By definition of l_P, we see that $l_P(A) = \min\{\alpha \mid \alpha > l_P(B), B \in \mathcal{B}_A\} \leq \min\{\alpha \mid \alpha > l(B), B \in \mathcal{B}_A\} \leq l(A)$. From this we obtain $l_P(A) \leq l(A)$, giving the required contradiction. ■

Application of the Prieß-Crampe and Ribenboim Theorem

We regard I_P as a domain, under set inclusion, whose set of compact elements is the set I_c of all finite subsets of B_P, see Section 3.2.

5.1.5 Definition Let P be a normal logic program and let $l : B_P \to \gamma$ be a level mapping. We define the rank function r_l *induced by* l by setting $r_l(I) = \max\{l(A) \mid A \in I\}$ for every $I \in I_c$, with I non-empty, and taking $r_l(\emptyset) = 0$. The generalized ultrametric obtained from a rank function in this way, see Definition 3.2.1, will be denoted by d_l and called the gum *induced by* l.

Note that d_l is spherically complete by Theorem 3.2.4.

The following proposition will make it easier to calculate distances which depend on r_l. To simplify notation, define $\mathcal{L}_\alpha = \{A \in B_P \mid l(B_P) \quad \}$ for each ordinal α.

5.1.6 Proposition Let P be a normal logic program, let $l : B_P \to \gamma$ be a level mapping for P and let $I, J \in I_P$. Then $d_l(I,J) = \inf\{2^{-\alpha} \mid I \cap \mathcal{L}_\alpha = J \cap \mathcal{L}_\alpha\}$, i.e. $d_l(I,J) = 2^{-\alpha}$, where α is the least ordinal such that I and J differ on some atom of level α.

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

We note that we could have used the characterization in Proposition 5.1.6 in order to define d_l more directly. The generalized metric d_l is in fact fundamental for the remaining chapter and will be the basis for the definitions of the generalized metrics employed in the sequel.

Our main result in this section is the following theorem.

5.1.7 Theorem Let P be a normal logic program which is locally hierarchical with respect to a level mapping $l : B_P \to \gamma$. Then T_P is strictly contracting with respect to the generalized ultrametric d_l induced by l, and T_P has a unique fixed point and hence P has a unique supported model.

Proof: Let $I_1, I_2 \in I_P$ and suppose that $d_l(I_1, I_2) = 2^{-\alpha}$.
Case 1. $\alpha = 0$.
Let $A \in T_P(I_1)$ with $l(A) = 0$. Since P is locally hierarchical, A must be the head of a unit clause in ground(P). From this it follows that $A \in T_P(I_2)$ also. By the same argument, if $A \in T_P(I_2)$ with $l(A) = 0$, then $A \in T_P(I_1)$. Therefore, $T_P(I_1) \cap \mathcal{L}_1 = T_P(I_2) \cap \mathcal{L}_1$, and hence we have

$$d_l(T_P(I_1), T_P(I_2)) \leq 2^{-1} \quad 2^{-} \; = d_l(I_1, I_2)$$

as required.
Case 2. $\alpha > 0$.
In this case, 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 $k_1, l_1 \geq 0$, such that for all k, j we have $A_k \in I_1$ and $B_j \in I_1$. Since P is locally hierarchical and $I_1 \cap \mathcal{L}_\alpha = I_2 \cap \mathcal{L}_\alpha$, it follows that for all k, j we have $A_k \in I_2$ and $B_j \in I_2$. Therefore, $A \in T_P(I_2)$. By the same argument, if $A \in T_P(I_2)$ with $l(A) \leq \alpha$, then $A \in T_P(I_1)$. Hence we have $T_P(I_1) \cap \mathcal{L}_{\alpha+1} = T_P(I_2) \cap \mathcal{L}_{\alpha+1}$, and it follows that

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

as required.

Thus, T_P is strictly contracting. Therefore, by the Prieß-Crampe and Ribenboim theorem 1.3.4, T_P has a unique fixed point and therefore P has a unique supported model as claimed. ■

In the case that l is an ω-level mapping, d_l is a conventional ultrametric and the Banach contraction mapping theorem 1.2.2 can be applied analogously to Theorem 5.1.7.

5.1.8 Theorem Suppose P is acyclic with level mapping l. Then T_P is a contraction with respect to the ultrametric d_l with contractivity factor $\frac{1}{2}$. Therefore, T_P has a unique fixed point by the Banach contraction mapping theorem 1.2.2, and hence P has a unique supported model.

We note also that it was shown in [Sed97] that the conventional ultrametric d_l, for an ω-level mapping l, generates the atomic topology on I_P in the case that $\mathcal{L}_n$ is finite for each $n \in \omega$. If this finiteness condition is not imposed on the level mapping, then the topology generated by d_l is finer than the atomic topology, which means that the sequence $(T_P^n(I))$, for each $I \in I_P$, which converges in d_l to the unique supported model of P by the proof of the Banach contraction mapping theorem 1.2.2 and Theorem 5.1.8, also converges with respect to the atomic topology.

In the case of a locally hierarchical program P, we can obtain a similar result by considering ordinal powers of T_P by setting $T_P^\alpha(I)$, for each limit ordinal α to be the set of all $A \in B_P$ such that A is eventually in $\left(T_P^\beta(I)\right)_{\beta<\alpha}$, and obtain, by the alternative proof of the Prieß-Crampe and Ribenboim theorem given in Theorem 1.3.9, that the transfinite sequence consisting of the ordinal powers of T_P at any given $I \in I_P$ converges in Q to the unique supported model of P; in fact this follows easily from the fact given in the proof of Theorem 1.3.9 that the transfinite sequence (T_P^α) is pseudo-convergent with respect to d_l, and that $d_l\left(T_P^\alpha, T_P^{\alpha+1}\right)$ is strictly decreasing and eventually 0 for increasing α.

Computational Adequacy of Locally Hierarchical Programs

We will show next that every partial recursive function can be implemented by a locally hierarchical program with cuts, and we will return to this in Chapter 6 from a different perspective. For details about SLDNF-resolution and about cuts, see [Llo88].

For convenience, we establish the following notation for every locally hierarchical program P. For $A \in B_P$, we say that $P \models A$ if and only if $A \in M_P$. We say that $P \vdash_{\text{SLDNF}} A$ if and only if there is an SLDNF-derivation for $P \quad \leftarrow A\}$. Recall that an SLDNF-derivation *flounders* [AP93] if a non-ground negative literal is selected at some stage in the derivation.

5.1.9 Theorem Let P be a locally hierarchical program and let $A \in B_P$ with $P \vdash_{\text{SLDNF}} A$. Then $P \models A$. If $\gamma_P = \omega$, and the SLDNF-derivation of $P \quad \leftarrow A\}$ does not flounder, then $P \vdash_{\text{SLDNF}} A$ if and only if $P \models A$. In particular, if P is without local variables, then $P \models A$ if and only if $P \vdash_{\text{SLDNF}} A$.

Proof: By [Llo88, Proposition 14.2], M_P is the unique model of the Clark completion comp(P) [Cla78, ABW88] of P. By [Llo88, Theorem 15.4], the first statement immediately holds. Now let $\gamma_P = \omega$ and $P \models A$ be such that the SLDNF-derivation of $P \quad \leftarrow A\}$ does not flounder. Then, by [AP93, Corollay 4.11], all $\cup\{$ SLDNF-derivations of $P \quad \leftarrow A\}$ are finite and, therefore, $P \vdash_{\text{SLDNF}} A$ which proves the second statement. If P is without local variables, then P is acyclic by Proposition 5.1.2 and obviously does not flounder on any ground goal, which completes the proof using the second statement. ■

We establish next the result that every partial recursive function can be computed by a locally hierarchical program with cuts. We take the point of view (following [Llo88]) that a cut does not affect the declarative semantics of a program. When talking about SLDNF-resolution for locally hierarchical programs with cuts, we assume that the selection function always selects the leftmost literal and, as discussed in [Llo88], that the cut prunes" the search tree. To obtain a well-defined procedural semantics of a given program, we assume that the topmost clause whose head unifies with a current goal is always selected first, as implemented in standard Prolog systems. So, for what follows, SLDNF-resolution is performed in the way just described.

For convenience, we will denote ground terms by lowercase letters and variables by uppercase letters when refering to a predicate. Thus, $p(x_1, \ldots, x_n, Y)$ means that all x_i are ground and Y is a variable. We write $(P, A) \vdash_{\mathrm{SLDNF}} B$ if $P \cup \{\leftarrow A\}$ has an answer substitution θ (via SLDNF-resolution) such that $A\theta = B$.

5.1.10 Theorem Identify $\mathbb{N}$ with the set of terms $\{s^n(0) \mid n \in \mathbb{N}\}$ by identifying s with the successor function. Let f be an n-ary partial recursive function. Then there exists a locally hierarchical program P_f with cuts and an $(n\ \ 1)$-ary predicate symbol p_f such that the following hold:

1. A call to P_f with goal $p_f(x_1, \ldots, x_n, Y)$ or $p_f(x_1, \ldots, x_n, y)$ terminates via SLDNF-resolution if $(x_1, \ldots, x_n) \in \mathrm{dom}(f)$ and backtracking over the goal fails immediately.

2. $(P_f, p_f(x_1, \ldots, x_n, Y)) \vdash_{\mathrm{SLDNF}} p_f(x_1, \ldots, x_n, y)$ if and only if $(x_1, \ldots, x_n) \in \mathrm{dom}(f)$ and $f(x_1, \ldots, x_n) = y$.

3. For every $p_f(x_1, \ldots, x_n, y) \in B_P$ the following are equivalent:

   (a) $P \models p(x_1, \ldots, x_n, y)$

   (b) $P \vdash_{\mathrm{SLDNF}} p(x_1, \ldots, x_n, y)$

   (c) $f(x_1, \ldots, x_n) = y$.

Proof: We follow [ŠŠ82] and [Llo88] with modifications where necessary. The proof is by induction on the number q of applications of composition, primitive recursion, and minimalization needed to define f.

Suppose first that $q = 0$. Thus f must be either the zero function, the successor function, or a projection function.

Zero function

Suppose that f is the zero function defined by $f(x) = 0$. Define P_f to be the program $p_f(X, 0) \leftarrow$.

Successor function

Suppose that f is the successor function defined by $f(x) = x$. Define P_f to be the program $p_f(X, s(X)) \leftarrow$.

Projection function

Suppose that f is the projection function defined by $f(x_1, \ldots, x_n) = x_j$ for some $j \in \{1, \ldots, n\}$. Define P_j to be the program $p_f(X_1, \ldots, X_n, X_j) \leftarrow$.

Clearly, for each of the basic functions, the program P_f, as defined, is locally hierarchical with the desired properties.

Next, suppose that the partial recursive function f is defined by $q > 0$ applications of composition, primitive recursion, and minimalization.

Composition

Suppose that f is defined by

$$f(x_1, \ldots, x_n) = h(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$$

where $g_1, \ldots, g_m$ and h are partial recursive functions. By the induction hypothesis, corresponding to each g_i (or h), there is a locally hierarchical program P_{g_i} (P_h) with cuts and a predicate symbol p_{g_i} (p_h) satisfying the conclusions of the theorem. We can suppose that the programs $P_{g_1}, \ldots, P_{g_m}, P_h$ do not have any predicate symbols in common. Define P_f to be the union of these programs together with the clause

$$\begin{aligned} p_f(X_1, \ldots, X_n, Z) \leftarrow & \, p_{g_1}(X_1, \ldots, X_n, Y_1), \ldots, p_{g_m}(X_1, \ldots, X_n, Y_m), \\ & h(Y_1, \ldots, Y_m, Z), !. \end{aligned}$$

Obviously, P_f is a locally hierarchical program with cuts. Statement 1 is immediate under the assertion of the induction hypothesis, as is the 'if'-part of statement 2. The 'only-if' part is shown as in [Llo88]. For statement 3, the equivalence of 3a and 3c is immediate and the equivalence of 3b and 3c is shown in a manner analogous to that employed in [ŠŠ82].

Primitive recursion

Suppose that f is defined by

$$\begin{aligned} f(x_1, \ldots, x_n, 0) &= h(x_1, \ldots, x_n) \\ f(x_1, \ldots, x_n, y+1) &= g(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y)) \end{aligned}$$

where h and g are partial recursive functions. By the induction hypothesis, corresponding to h (resp. g), there is a locally hierarchical program P_h (resp. P_g) with cuts and a predicate symbol p_h (resp. p_g) satisfying the conclusions of the theorem. We can also suppose that P_h and P_g do not have any predicate symbols in common. Define P_f to be the union of P_h and P_g together with the clauses

$$\begin{aligned} p_f(X_1, \ldots, X_n, 0, Z) &\leftarrow p_h(X_1, \ldots, X_n, Z), !. \\ p_f(X_1, \ldots, X_n, s(Y), Z) &\leftarrow p_f(X_1, \ldots, X_n, Y, U), p_g(X_1, \ldots, X_n, Y, U, Z), !. \end{aligned}$$

Obviously, P_f is a locally hierarchical program with cuts. The desired properties are proven along the same lines as for composition.

Minimalization

Suppose that f is defined by $f(x_1, \ldots, x_n) = \mu y(g(x_1, \ldots, x_n, y) = 0)$ where g is a partial recursive function. By the induction hypothesis, corresponding to

g there is a locally hierarchical program P_g with cuts and a predicate symbol p_g satisfying the conclusions of the theorem. Define P_f to be P_g together with the clauses

$$
\begin{aligned}
p_f(X_1, \ldots, X_n, 0) &\leftarrow p_g(X_1, \ldots, X_n, 0, 0), !. \\
p_f(X_1, \ldots, X_n, s(\quad)) &\leftarrow r(X_1, \ldots, X_n, Z\,), p_g(X_1, \ldots, X_n, s(\quad), 0), !. \\
r(X_1, \ldots, X_n, 0) &\leftarrow \neg p_g(X_1, \ldots, X_n, 0, 0). \\
r(X_1, \ldots, X_n, s(\quad)) &\leftarrow r(X_1, \ldots, X_n, Z\,), \neg p_g(X_1, \ldots, X_n, s(\quad), 0).
\end{aligned}
$$

Obviously, P_f is a locally hierarchical program with cuts. Again, statements 1 and 2 are proven along the same lines as for composition by taking into account the fact that, if p_g occurs in a subgoal of the computation, it is always ground. Note that $r(x_1, \ldots, x_n, z) \in M_{P_f}$ if and only if $(x_1, \ldots, x_n, k) \in \mathrm{dom}(g)$ and $g(x_1, \ldots, x_n, k) = 0$ for every $k \quad z$, and that the goal $r(x_1, \ldots, x_n, Z\,)$ subsequently yields all answer substitutions $\quad z$ $(Z/ = 0, 1, 2, \ldots)$ with $(x_1, \ldots, x_n, k) \in \mathrm{dom}(g)$ and $g(x_1, \ldots, x_n, k) = 0$ for all $k \quad$, which yields the equivalence of 3b and 3c. To show the equivalence of 3a and 3c, note that $P \models r(x_1, \ldots, x_n, z)$ if and only if $P \models p_g(x_1, \ldots, x_n, k, 0)$ for all $k \quad$. So $P \models p_f(x_1, \ldots, x_n, z)$ if and only if $P \models p_g(x_1, \ldots, x_n, z, 0)$ and $P \models p_g(x_1, \ldots, x_n, k, 0)$ for all $k \quad z$. Now suppose $f(x_1, \ldots, x_n) = z$. Then by the induction hypothesis, the above yields that $P \models p_f(x_1, \ldots, x_n, z)$. Now suppose $f(x_1, \ldots, x_n) = z$. We consider three cases:
(1) $g(x_1, \ldots, x_n, z) = 0$. Then $P \models p_f(x_1, \ldots, x_n, z)$ immediately.
(2) $g(x_1, \ldots, x_n, k) = 0$ for some $k < \ z$. Again $P \models p_f(x_1, \ldots, x_n, z)$ immediately.
(3) $(x_1, \ldots, x_n, k) \in \mathrm{dom}(g)$ for some $k \quad z$. Then $r(x_1, \ldots, x_n, k)$ occurs as a subgoal of the computation and, therefore, so does $p_g(x_1, \ldots, x_n, k, 0)$. Note that g cannot be one of the basic functions since they are total. For the same reason, g cannot be defined by using composition and primitive recursion on the basic functions only. Consequently, at some point in the computation, a subgoal $p_{f_0}(x_1, \ldots, x_n, y)$ or $p_{f_0}(x_1, \ldots, x_n, Y)$ occurs with $f\ (x_1, \ldots, x_n) = \mu y(g\ (x_1, \ldots, x_n, y) = 0)$ and $(x_1, \ldots, x_n) \in \mathrm{dom}(f\ )$. There are two subcases to consider:
(i) $g\ (x_1, \ldots, x_n, m) = 0$ for all $m \in \mathbb{N}$. It is easily seen that in this case P_{f_0} will not terminate on the subgoal $p_{f_0}(x_1, \ldots, x_n, Y)$ and will fail on the subgoal $p_{f_0}(x_1, \ldots, x_n, y)$.
(ii) $(x_1, \ldots, x_n, m) \in \mathrm{dom}(g\ )$ for some $m \in \mathbb{N}$. The condition of this case is exactly as in case (3).
Thus, the argument can be repeated. Since every partial recursive function is defined by using minimalization only finitely often, the conclusion follows by induction. ■

Theorem 5.1.10 shows that locally hierarchical programs with cuts are computationally adequate with respect to SLDNF-resolution as interpreter. We note that the cuts occurring in the proof are *safe* in the sense that they cut only branches of the search tree which do not contain any success branches.

5.2 Acceptable Programs

Acceptable programs were first studied in detail in [AP93] where they were shown to coincide, basically, with the programs which are left-terminating. In [AP94, Apt95], they were further examined in the context of formal verification under Prolog. The acceptable programs therefore form an important class. However, in order to show from the definition that a given program P is acceptable, it is necessary to determine a level mapping and a model for P which satisfy the conditions of the definition, see Definition 5.0.2. But this may be difficult to do, and it is therefore desirable to simplify this task, if possible, and we will now take some steps in this simplification process by shedding light on the behaviour of the single-step operator in this case.

Most of the methods and results in this section can easily be carried over to the more general classes of programs which will be studied in the remaining part of the chapter. We have decided to present them for the more special case of acceptable programs due to the importance of this class of programs.

A first attempt at studying acceptable programs from a topological perspective was made in [Fit94]. In this paper, a distance function d_3 associated with a given acceptable program was defined, which acts on I_P. This distance function turns out to be a dislocated metric, and our approach builds heavily on this distance function, showing that it can be put to good use for studying, and characterizing, acceptability.

The single-step operator T_P is in fact a contraction with respect to d_3 if P is acceptable, and we will see that convergence of iterates of T_P in the atomic topology follows from this, and the limit M_P of the sequence of iterates of T_P will be seen to be the unique supported model of P (Theorem 5.2.10). The existence of a unique supported model of an acceptable program was already established in [AP93], in the case of Herbrand preinterpretations. It was obtained as the supremum of the iterates of the monotonic three-valued operator Φ_P from [Fit85], cf. Chapter 6. Our characterization by means of T_P and Q simplifies this process since the single-step operator is easier and more natural to apply.

The topological characterization of M_P just described, will also easily allow us to establish the fact that a program P, which is acceptable with respect to some model I and level mapping l, is also acceptable with respect to M_P and l (Theorem 5.2.12). Even more, we will show that M_P is the smallest of all models with respect to which acceptability of P can be established (Corollary 5.2.13).

At this stage, we know that convergence in Q of iterates of T_P is a neccessary condition for acceptability of P. If this condition is met, the limit M_P thus obtained is suitable for establishing acceptability if a corresponding level-mapping is found. And in fact, every level mapping which renders P acceptable with respect to some model, will also allow one to establish acceptability of P with respect to M_P (Theorem 5.2.12). The set of all these possible level mappings will finally turn out to contain a pointwise least element (Theorem 5.2.21). For this level mapping, which will be called the canonical acceptable-level mapping l_P for P, we will give an iterative construction, provided M_P is known (Construction

5.2.16). This construction, in fact, is applicable to all programs and depends on a given model of the program. In this general case, however, the construction may only lead to a partial mapping. From this, again, we derive a necessary condition for acceptability of P, namely that the construction of l_P, using the model M_P, yields a level mapping which is not partial (Proposition 5.2.17).

The iterative methods for obtaining M_P and l_P then provide a means for characterizing, and establishing, acceptability of a program in question. This is done by subsequently conducting the following steps (Theorem 5.2.19). (1) Obtain iterates of T_P. If they converge in Q, call the limit M_P. If they don't converge, then P is not acceptable. (2) Obtain l_P using M_P. If l_P is not total, then P is not acceptable. (3) Check whether condition (5.1) of Definition 5.0.2 holds. If it holds, then P is acceptable. If it does not hold, then P is not acceptable.

Conducting steps (1) and (2) above is by no means a trivial task and in fact is an undecidable problem. Our characterization, however, sheds more light on the concept of acceptability and might be an aid for determining acceptability if straightforward attempts fail. Simplification of this process is achieved by a result which allows to partition the program in question into subprograms in a way that subsequent establishment of acceptability of the subprograms suffices for determining acceptability (Lemma 5.2.25 and Theorem 5.2.26).

Finally, the results obtained will be applied in order to show that both M_P and l_P are suitable for establishing termination of general non-ground queries.

In order to simplify notation in this section, we will abbreviate $\mathtt{Neg}_P^*$ by N.

Remarks on Domains of Preinterpretation

The choice of a suitable domain of preinterpretation is essential in the sense that a program might be acceptable under some chosen domain, and not be acceptable under another. We will illustrate this and the difficulties involved by means of a few example programs.

5.2.1 Program Let P_1 be the following program.

$$
\begin{aligned}
&r(0) \leftarrow \neg p(0), \neg r(0)\\
&p(0) \leftarrow \neg q(X)\\
&q(0) \leftarrow
\end{aligned}
$$

Here, $P_1^- = P_1$ and P is acceptable with respect to the supported model $\{p(0), q(0)\}$, whose domain is the set $\{0, 1\}$, and the level mapping given by $l(q(0)) = l(q(1)) = 0$, $l(p(0)) = l(p(1)) = 1$, $l(r(0)) = l(r(1)) = 2$. However, P fails to have any supported models if the domain of preinterpretation contains only the constant and function symbols occurring in the program.

5.2.2 Program Let P_2 be the following program.

$$
\begin{aligned}
&r(0) \leftarrow \neg q(X), \neg r(0)\\
&q(0) \leftarrow
\end{aligned}
$$

The program P_2 is acceptable with respect to the domain $\{0\}$. However, it has no supported model with respect to the set $\{0, 1\}$ as domain of preinterpretation. Note that the programs P_1 and P_2 flounder on some goals.

Constructive negation in the sense of [Cha88] (cf. also [Mar96]), as a way to resolve floundering, does not cover the general case either, due to the following two assumptions made in the cited papers: Chan in [Cha88, p. 113] assumes, throughout, the consistency of the completed database, and also assumes [Cha88, p. 116] that the underlying language (i.e. the domain of preinterpretation) contains infinitely many constant symbols and function symbols.

Consistency of the completed database is dependent on the chosen domain of preinterpretation (restricted here through the presence of infinitely many constant and function symbols) and, in fact, under the assumption concerning the underlying language as above, we see that the completed database for program P_2 is not consistent.

Furthermore, consider the following program.

5.2.3 Program Let P_3 be the following program.

$$\begin{aligned} r(0) &\leftarrow \neg q(X), r(0) \\ q(0) &\leftarrow \end{aligned}$$

For program P_3, the unique supported Herbrand model $\{q(0)\}$ is certainly the desired model. The program is also acceptable with respect to this model.

However, the goal $\leftarrow r(0)$, which is bounded, does not terminate under Chan's constructive negation. In [Mar96], however, it was shown that the set of all programs which are acceptable with respect to some preinterpretation J whose domain contains infinitely many constants and functions, coincides with the set of all programs which terminate under Chan's constructive negation. Nevertheless, the result does not account for programs which are acceptable with respect to a domain containing finitely many constants and functions, but not with respect to a domain which is constrained as for constructive negation. The Program P_3 displays this fact.

In all previous examples, the Herbrand preinterpretation was too *small* to allow determination of acceptability. Our final program shows that in some cases it may even be too *large*.

5.2.4 Program Let P_4 be the following program.

$$\begin{aligned} r(0) &\leftarrow \neg q(X), r(0) \\ q(f(0)) &\leftarrow \end{aligned}$$

Under the domain $\{f^n(0) \mid n \in \mathbb{N}\}$, this program is not acceptable due to the existence of the function symbol f, giving an instance of $q(X)$ which is false. However, P_4 is acceptable with respect to a preinterpretation whose domain is the one-point set $\{0\}$ and where f is interpreted as the identity function on $\{0\}$.

In fact, this example shows that the result from [AP93] which states that every program which is acceptable with respect to a Herbrand preinterpretation has a unique supported Herbrand model, cannot be generalized to arbitrary preinterpretations in general.

On the other hand, [AP93, Corollary 4.12] shows that every acceptable program is left terminating, whilst [AP93, Theorem 4.18] contains the result that every left terminating non-floundering program is acceptable. Moreover, the proof given of this latter fact shows that one has acceptability with respect to some Herbrand model, where the underlying domain of preinterpretation is constructed using only the variable and constant symbols occurring in the program   for such programs, we suggest the terminology *Herbrand-acceptable*. Thus, an acceptable program which fails to be Herbrand-acceptable must flounder on some ground query. Moreover, all the examples considered in [AP93] are Herbrand-acceptable.

In the following, as already noted, we will work over arbitrary preinterpretations.

Fitting's Approach

As already noted, it was first shown in [AP93] that every (Herbrand-) acceptable program has a unique supported model. In [Fit94], Fitting considered proving the same result by using metrics and the Banach contraction mapping theorem. His method depends on the following definitions. A *partial level mapping* is a partial mapping $l : B_P \to \alpha$, where α is an ordinal. Recall the notation $\mathcal{L}_\beta$ for the set of all atoms A of level $l(A)$ less than β. For the remainder of this section, we will consider only ω-level mappings, i.e. $\alpha = \mathbb{N}$.

5.2.5 Definition Let P be a normal logic program with partial level mapping l. The pseudometric d associated with l on I_P is defined as follows. For $J, K \in I_P$ let

$$d(J,K) = \inf\{2^{-n} \mid \mathcal{L}_m \cap \mathrm{dom}(l) \cap J = \mathcal{L}_m \cap \mathrm{dom}(l) \cap \quad \text{for all } m \leq n\},$$

where $\mathcal{L}_m$, for all $m \in \mathbb{N}$, is taken with respect to a (total) level mapping l' which extends l.

By [Fit94], any pseudometric associated with a (partial) level mapping is complete.

If the level mapping is *total*, i.e. not a partial mapping, Definition 5.2.5 coincides with the metric d_l of Proposition 5.1.6.

5.2.6 Definition Let P be acceptable with respect to a level mapping l and a model I. We define the partial level mappings l_1 and l_2 as follows; recall that we write N instead of $\mathtt{Neg}_P^*$.

1. $\mathrm{dom}(l_1) = N$, $l_1(A) = l(A)$ for all ground literals A in N.

2. $\mathrm{dom}(l_2) = {}^cN$, $l_2(A) = l(A)$ for all ground literals A not in N.

The associated pseudometrics are denoted by d_1 and d_2, respectively. Furthermore, we define a function $\rho : I_P \to \mathbb{R}$ by

$$\rho(J) = \inf\{2^{-n} \mid J \cap {}^cN \cap \mathcal{L}_n \quad I\}.$$

This form of ρ differs only slightly from that used in [Fit94] and can easily be shown to be equivalent. Finally, following [Fit94] again, we define for all $J,K\ \in I_P$

$$d_3(J,K\ ) = \max\{d_1(J, I), d_1(\quad I)Kd_2(J,K\ ), \rho(J), \rho(\quad)\}. \tag{5.2}$$

We note that this distance function d_3 depends both on the level mapping l and on the interpretation I. We will discuss the intuition behind the definition of d_3 after Proposition 5.2.8, which will provide us with some understanding of this distance function. For the moment, we note that d_3 is a dislocated metric, but that it is not in fact a metric. Indeed, let P be the program consisting of the three unit clauses $p(0) \leftarrow, q(0) \leftarrow, q(1) \leftarrow$, where 0 and 1 are constant symbols. Then P is acceptable with respect to the Herbrand model $I = \{p(0), q(0), q(1)\}$ and the zero level mapping l. A straightforward calculation shows that $d_3(J, {}^cI) = 1$ for all $J \in I_P$ so that, in particular, one has $d_3({}^cI, {}^cI) = 1$. Nevertheless, it will turn out to be a useful tool in formulating some of our results. In fact, the following proposition, [Fit94, Proposition 7.1], does not need the assumption that d_3 is a metric and will be useful later.

5.2.7 Proposition Let P be acceptable with respect to a level mapping l and a model I. Then for all $J,K\ \in I_P$ we have $d_3(T_P(J), T_P(\quad)) \leq \frac{1}{2} d_3(J,K\ )$.

Applying the Matthews Theorem

We start by examining the relationship between the atomic topology Q and Fitting's dislocated metric d_3. The following result will clarify the behaviour of sequences which converge in d_3.

5.2.8 Proposition Let P be acceptable with respect to a level mapping l and a model I. Let J_n be a sequence which converges in d_3 to some $J \in I_P$. Then the sequence J_n converges to J in Q, and the following two conditions hold.

(i) $J_n \cap N$ converges in Q to the model $I \cap N$ of $\mathrm{comp}(P^-)$.

(ii) $J_n \cap {}^cN$ converges in Q to some $\qquad I$.

Furthermore, we obtain $J = (I \cap N) \qquad .$

Proof: By hypothesis, we have $d_3(J_n, J) \to 0$ as $n \to \quad$. By definition of d_3 this implies that $d_1(J_n, I)$, $d_1(J, I)$ and $d_2(J_n, J)$ all tend to 0 as $n \to \quad$. Hence, by definition of d_1 and d_2, it follows that for all $m \in \mathbb{N}$ there exists some $n \quad \in \mathbb{N}$ such that for all $n \geq n \quad$ we have

$$\begin{aligned} J_n \cap N \cap \mathcal{L}_m &= I \cap N \cap \mathcal{L}_m, \\ J \cap N \cap \mathcal{L}_m &= I \cap N \cap \mathcal{L}_m \qquad \text{and} \\ J_n \cap {}^cN \cap \mathcal{L}_m &= J \cap {}^cN \cap \mathcal{L}_m. \end{aligned}$$

From these equations, it follows that for all $m \in \mathbb{N}$ there exists some $n \ \in \mathbb{N}$ such that for all $n \geq n$ we have $J_n \cap \mathcal{L}_m = J \cap \mathcal{L}_m$ which proves convergence of the sequence J_n to J in Q.

We also obtain that $J_n \cap N$ and $J_n \cap {}^cN$ converge in Q to $J \cap N$ respectively $J \cap {}^cN$. By definition of d_3 we have $d_1(J, I) = 0$ which implies that $J \cap N = I \cap N$. From the same definition we obtain $\rho(J) = 0$ and therefore $\quad = J \cap {}^cN \quad I$ which completes the proof. ■

As a corollary from the proof of Proposition 5.2.8, we obtain that convergence in d_3 is independent of the choice of level mapping.

We are now in a position to better understand the intuition underlying the definition of d_3 given in equation (5.2). Essentially, the terms $d_1(J, I)$ and $d_1(\quad I)K$, in this equation ensure that if $d_3(J,K)$ is small, then both J and are "close" (with respect to the pseudometric d_1) to the chosen interpretation I, and this closeness depends only on the atoms contained in N. Convergence in d_3 means that the sequence in question must tend towards the unique supported model $I \cap N$ of P^-. The remainder of the definition constrains what "closeness" means on ${}^cN$. The term $d_2(\quad J)K$, ensures that and J share "enough" elements (of suitable level), and the ρ-function forces both and J to be largely a subset of I on ${}^cN$. In terms of convergence in d_3, the distance function d_3 could be understood as "filtering" a sequence towards a suitable subset of I, namely a subset which coincides with I on N.

5.2.9 Proposition The d-metric d_3 is complete.

Proof: Let J_n be a Cauchy sequence with respect to d_3. By definition of d_3, this implies that $d_1(J_{m_1}, I)$, $d_1(J_{m_2}, I)$, $d_2(J_{m_1}, J_{m_2})$, $\rho(J_{m_1})$ and $\rho(J_{m_2})$ all tend to 0 for $m_1, m_2 > m$ and increasing m, and we obtain, as in the proof of Proposition 5.2.8, that J_n converges in Q to some J. An argument similar to that in the proof of Proposition 5.2.8 again shows that J is also the limit of J_n with respect to d_3. ■

5.2.10 Theorem Let P be acceptable with respect to a level mapping l and a model I, and let $\quad \in I_P$ be arbitrary. Then $T_P^n(\quad)$ converges in Q to the unique supported model M_P of P.

Proof: The d-metric d_3 is complete by Proposition 5.2.9, and T_P is a contraction with respect to d_3 by Proposition 5.2.7. So we can apply the Matthews theorem 1.4.6, which yields that the sequence $T_P^n(\quad)$ converges in d_3 to the unique supported model of P. Since convergence in d_3 implies convergence in Q by Proposition 5.2.8, the proof is complete. ■

Minimality of the Unique Supported Model

We will now provide an alternative characterization of the model M_P. Recall that we are working under a fixed but arbitrary preinterpretation.

5.2.11 Lemma Let P be acceptable and let $\mathcal{I}$ be the set of all models with respect to which P can be established to be acceptable. Then $M_P \cap N = I \cap N$ for all $I \in \mathcal{I}$. In particular, $I \cap N = J \cap N$ for all $I, J \in \mathcal{I}$. Furthermore, we have the minimality property $M_P \quad \mathcal{I}$.

Proof: The sequence $J_n = T_P^n(\emptyset)$ converges with respect to d_3 and satisfies conditions (i) and (ii) of Proposition 5.2.8 for all $I \in \mathcal{I}$. The first statement follows then immediately from condition (i) and the second statement from condition (ii). ■

The model thereby obtained will be shown to be suitable for demonstrating the acceptability of the program in question. We will need this result for our characterization of acceptability in Theorem 5.2.19, and it will also give us an alternative characterization of M_P as an easy corollary.

5.2.12 Theorem Let P be acceptable with respect to a level mapping l and a model I. Then P is acceptable with respect to l and M_P.

Proof: Since $I \cap N = M_P \cap N$ by Lemma 5.2.11, it remains to show that the acceptability condition (5.1) from Definition 5.0.2 holds. Again by the same result, it remains to show the condition for all clauses which are not in P^-. Since $M_P \cap N = I \cap N$, and therefore these agree on all ground atoms which occur negatively in P, it suffices to show that $M_P \quad I$, which is the case by Lemma 5.2.11. ■

5.2.13 Corollary Let P be acceptable and let $\mathcal{I}$ be the set of all models with respect to which P can be established to be acceptable. Then $M_P = \quad \mathcal{I}$.

Proof: This follows immediately from Lemma 5.2.11 and Theorem 5.2.12. ■

The Canonical Level Mapping for Acceptable Programs

We show next how to obtain a level mapping for a given program which is suitable for proving its acceptability. The construction is based on Construction 5.1.1 for locally hierarchical programs. For this purpose, let P be a program and I a model of P. We will now give a program transformation which yields a locally hierarchical program from P and I if P is acceptable with respect to I, allowing us to apply our earlier results. The program transformation is as follows:

5.2.14 Program Transformation Let P be a normal logic program and I a model of P. For each clause $A \leftarrow L_1, \ldots L_n$ in ground(P) determine the maximal i such that $I \models L_1 \quad L_i$. Then replace the given clause with $A \leftarrow L_1, \ldots, L_{i+1}$ if $i = n$ and by $A \leftarrow L_1, \ldots, L_n$ if $i = n$. The resulting ground program will be called P_I.

If P is acceptable with respect to I and l, then P_I is locally hierarchical with respect to the ω-level mapping l' which is obtained by restricting l to B_{P_I}.

Therefore, we can obtain the canonical lh-level mapping l_{P_I} of P_I by applying Construction 5.1.1, and obtain by Corollary 5.1.3 that l_{P_I} is indeed a total function. Furthermore, by Proposition 5.1.4 we obtain that $l_{P_I}(A) \leq l'(A)$ for all $A \in B_{P_I}$, and since l' maps into ω, the level mapping l_{P_I} also maps into ω. This means, in particular, that Construction 5.1.1 is in fact not transfinite but closes off at ω.

5.2.15 Definition We now define a level mapping l_P for the given program P: For every $A \in B_P \setminus B_{P_I}$ let $l_P(A) = 0$. For every $A \in B_{P_I}$ let $l_P(A) = l_{P_I}(A)$.

We summarize the observations just discussed.

5.2.16 Construction Let P be a normal logic program and I a model of P.

(1) Obtain P_I from P and I using Program Transformation 5.2.14.

(2) Obtain l_{P_I} from Construction 5.1.1.

(3) Obtain l_P from Definition 5.2.15.

5.2.17 Proposition Let P be acceptable with respect to a model I. Then the following statements hold.

(i) P_I, obtained from step (1) in Construction 5.2.16 is locally hierarchical.

(ii) l_{P_I}, obtained from step (2) in Construction 5.2.16 is total (with respect to B_{P_I}) and maps into ω.

(iii) l_P, obtained from step (3) in Construction 5.2.16 is total and maps into ω.

(iv) P is acceptable with respect to I and l_P.

Proof: It only remains to prove statement (iv), which is immediate from the definition of l_P. ■

In the following, l_P will also denote the (partial) level mapping as given in Construction 5.2.16. It will be called the *canonical (partial) acceptable-level mapping* for P.

The following is the key result in our characterization of acceptability.

5.2.18 Theorem Let P be acceptable. Then P is acceptable with respect to M_P and l_P.

Proof: By Theorem 5.2.12, P is acceptable with respect to l and M_P. By Proposition 5.2.17, P is then acceptable with respect to l_P and M_P. ■

We can now state the following characterization theorem.

5.2.19 Theorem Let P be a normal logic program. Then P is acceptable if and only if the following conditions are satisfied:

(1) The sequence $(T_P^n(\emptyset))_{n\in\mathbb{N}}$ converges in Q to some M_P.

(2) The mapping l_P, constructed from P and $I = M_P$ as in Construction 5.2.16, is total and takes values in the natural numbers.

(3) P satisfies condition (5.1) from Definition 5.0.2 with respect to l_P and M_P.

Proof: Let P be acceptable. Then (1) follows from Theorem 5.2.10, (2) follows from Proposition 5.2.17, and (3) follows from Theorem 5.2.18. The converse is immediate. ■

Minimality Properties

We show that the canonical acceptable-level mapping l_P of P is least among all level mappings with respect to which acceptability can be established.

5.2.20 Lemma Let P be acceptable with respect to M_P and some level-mapping l. Then $l_P(A) \leq l(A)$ for all $A \in B_P$.

Proof: For $A \in B_{P_{M_P}}$, we obtain $l_P(A) \leq l(A)$ by Proposition 5.1.4. If $A \in B_P \setminus B_{P_{M_P}}$, then by definition of l_P we have $l_P(A) = 0 \leq l(A)$ as desired. ■

5.2.21 Theorem For any acceptable program P, the canonical acceptable-level mapping l_P is least among all level mappings with respect to which P can be shown to be acceptable. More precisely, if P is acceptable with respect to some model I and some level mapping l, then for all $A \in B_P$ we have $l_P(A) \leq l(A)$.

Proof: Let P be acceptable with respect to some model I and some level mapping l, and let $A \in B_P$ be arbitrarily chosen. By Theorem 5.2.12, P is acceptable with respect to l and M_P. By Lemma 5.2.20 we obtain $l_P(A) \leq l(A)$ as desired. ■

Partitioning Acceptable Programs

In order to simplify the calculation of M_P, we will use methods similar to those employed in [ABW88, Prz88, Mar95]. We will use the following definition which is similar to [Mar95, Definition 4.1]. For any given progam P, recall that a predicate symbol p is said to be *defined* in a subprogram R of P if every clause which contains p in its head is contained in R. The *definition* of a predicate symbol is the smallest subprogram R such that the predicate symbol is defined in R. This notion extends naturally to atoms.

5.2.22 Definition Let P be a program and Q and R be two subprograms of P. We say that R *extends* Q, written $R > Q$, if no predicate symbol defined in R occurs in Q.

The basic idea is to partition an acceptable program in a suitable way such that M_P can be obtained by calculating the corresponding models of the subprograms in sequence.

5.2.23 Definition Let P be acceptable and $P = P_1 \quad \dots \quad P_k$. We call $(P_1, \dots, P_k)$ an *acceptable stratification* of P if $P_{i+1} > P_i$ for all $i = 1, \dots, k-1$.

By `true` and `false`, we will subsequently denote atoms which always evaluate to true and false, respectively. Now apply the following construction.

Replace every atom in each clause in ground(P_1) which does not occur in the head of any clause by `false`, and call the resulting program P'_1. By M_1, we will denote M_P restricted to the predicate symbols occurring in P_1, and by l_1 we will denote l_P restricted to the predicate symbols occurring in P_1. We obtain the following result.

5.2.24 Lemma Let P be acceptable with acceptable stratification $(P_1, \dots, P_k)$. Then the following hold.

(i) P'_1 is acceptable.

(ii) The sequence $T^n_{P_1}(\emptyset)$ of iterates converges in the atomic topology to the unique supported model M_1 of P'_1.

Proof: (i) P'_1 obviously is acceptable with respect to M_1 and l_1.
(ii) By Theorem 5.2.10, the iterates converge to a supported model of P'_1. By uniqueness of this model it coincides with M_1. ■

Let M_i, for $i = 1, \dots, k$, denote M_P restricted to the predicate symbols defined in P_i. Now suppose that for some $i \in \{1, \dots, k-1\}$ the programs $P'_1, \dots, P'_i$ have been defined and that the following properties have been established.

1. $P'_1, \dots, P'_i$ are acceptable.

2. M_i is the unique supported model of P'_i and $M_1 \qquad M_i$ is the unique supported model of $P_1 \qquad P_i$.

Then define P'_{i+1} by replacing all occurrences of atoms in ground(P_{i+1}) which are not defined in P_{i+1}, by `true` or `false`, respectively, depending on whether the atom is true or false, respectively, with respect to $M_1 \qquad M_i$. We then obtain the following result.

5.2.25 Lemma Suppose the assumptions above hold. Then the following hold.

(i) P'_{i+1} is acceptable.

(ii) The sequence $T^n_{P_{i+1}}(\emptyset)$ of iterates converges in the atomic topology to the unique supported model M_{i+1} of P'_{i+1}.

(iii) $M_1 \quad\quad M_{i+1}$ is the unique supported model of $P_1 \quad\quad P_{i+1}$.

Proof: (i) M_{i+1} is a supported model of P'_{i+1}, since M_P is a supported model of P and P'_{i+1} was obtained from P_{i+1} by replacing atoms with `true` or `false` according to their value with respect to the model $M_1 \quad\quad M_i$, and this coincides with M_P restricted to the predicate symbols defined in $P_1 \quad\quad P_{i+1}$. Therefore, P'_{i+1} is acceptable with respect to this model and l_P restricted to the predicate symbols in P'_{i+1}.
(ii) Convergence is again ensured by the acceptability of the program. Also, by Theorem 5.2.10, these iterates converge to the unique supported model of P'_{i+1} which is exactly M_{i+1} by the observations made in (i).
(iii) This is immediate by the assumption and (ii). ■

Putting all these results together, we obtain the following Theorem.

5.2.26 Theorem Let P be acceptable with acceptable stratification $(P_1, \ldots, P_k)$. For $i = 1, \ldots, k$ let M_i be constructed as above. Then $M_1 \quad\quad M_k = M_P$.

Termination of Non-Ground Queries

We cite the following result from [Apt95, Theorem 5.7]. For a partial converse, see [AP93].

5.2.27 Theorem Let P be acceptable with respect to a level mapping l and a model I. Then, for every literal L which is bounded with respect to l, all SLDNF-derivations of $P \cup \{\leftarrow L\}$, using the Prolog selection rule, are finite. In particular, the goal $\{\leftarrow L\}$ terminates under Prolog.

With our preparations, the following result is easily obtained.

5.2.28 Theorem Let P be acceptable with respect to a level mapping l and a model I, and let L be a literal which is bounded with respect to l. Then L is bounded with respect to l_P.

Proof: This follows immediately from the minimality of l_P as established in Theorem 5.2.21. ■

We will now discuss termination of non-ground, i.e. general, goals. The following notions were introduced in [AP93].

A *multiset* or *bag* over a set W is an unordered sequence of elements of W. Given a (non-reflexive) ordering on a set W, the *multiset ordering* over $(W,<)$ is an ordering of finite multisets of the set W and is defined as follows. For two finite multisets X and Y over W, let $X \prec Y$ if and only if $X = (Y \setminus \{a\})$ for some finite multiset such that $b \quad a$ for all $b \in$. Finally, define the multiset ordering over $(W,<)$ as the transitive closure of the relation $\prec$. The multiset whose elements are $a_1, \ldots, a_n$ will be denoted by $\mathrm{bag}(a_1, \ldots, a_n)$.

The following definition is to be found in [AP93, Definition 2.9].

5.2.29 Definition Let P be a program, l a level mapping for P, I a model of P with $I \cap N$ being a model for P^-, and let $k \geq 0$.

(i) With each ground goal G of the form $\leftarrow L_1, \ldots, L_n$ we associate a finite multiset $l_I(G)$ of natural numbers defined by $l_I(G) = \text{bag}(l(L_1), \ldots, l(L_{n(G,I)}))$, where $n(G, I) = \min(\{n\} \cup \ \{i \in \{1, \ldots, n\} \mid I \models L_i\})$.

(ii) With each goal G we associate a set of multisets $l'_I(G)$ defined by $l'_I(G) = \{l_I(G') \mid G'$ is a ground instance of $G\}$.

(iii) A goal G is called *bounded* by k with respect to l and I if $k \geq j$ for all $j \in \quad l'_I(G)$, where $\quad l'_I(G)$ stands for the set-theoretic union of the elements of $l'_I(G)$.

(iv) A goal is called *bounded* with respect to l and I if it is bounded by some $k \geq 0$ with respect to l and I.

It was observed in [Apt95] that the choice of level mapping and of the model can affect the class of (general, non-ground) goals whose termination can be established, since the choice of both the level mapping and the model affect the notion of boundedness for goals. However, we will prove that the model M_P and the canonical acceptable-level mapping l_P are completely general for proving termination of non-ground goals.

The following result is taken from [AP93, Corollary 4.11]. A partial converse is also given there.

5.2.30 Theorem Let P be an acceptable program and G a bounded goal. Then all SLDNF-derivations of $P \quad G\}$, using the Prolog selection rule, are finite. $\cup\{$

Our minimality results allow us to establish the following.

5.2.31 Theorem Let P be acceptable with respect to a level mapping l and a model I, and let G be a goal which is bounded with respect to l and I. Then G is bounded with respect to l_P and M_P.

Proof: Since $l_P(A) \leq l(A)$ for all $A \in B_P$ by Theorem 5.2.21, it suffices to show that $n(G, M_P) \leq n(G, I)$. This, however, follows directly from the minimality properties given in Lemma 5.2.11 and Theorem 5.2.21. ■

We note, finally, that the model M_P does not in general describe the procedural semantics of the program due to the possible presence of floundering intermediate goals, cf. [AP93] and [Apt95]. The exact relationship between M_P and the procedural semantics of P remains to be established.

5.3 Φ^*_ω-Accessible Programs

We associate a dislocated metric to each Φ^*-accessible program, show that it coincides with the d-metric d_3 from Section 5.2, and apply the Matthews theorem 1.4.6.

In the following, P is a Φ^*-accessible program which satisfies the defining conditions with respect to a model I and a level mapping l, see Definition 5.0.4.

For $J,K \in I_P$ we now define $d(\quad K) = 0$ and $d(J,K) = 2^{-n}$, where J and differ on some atom $A \in B_P$ of level n, but agree on all ground atoms of lower level, i.e. d coincides with the metric d_l induced by l. As was pointed out in [Fit94], and as we know from Theorem 3.2.4, (I_P, d) is a complete metric space, in fact even an ultrametric space. We also define a function $f : I_P \to \mathbb{R}$ by $f(\quad) = 0$ if $\quad I$ and $f(\quad) = 2^{-n}$, where n is the smallest integer such that there is an atom $A \in B_P$ with $l(A) = n$, $\quad \models A$ and $I \models A$. Finally, we define $u : I_P \to \mathbb{R}$ by $u(\quad) = \max\{f(\quad '), d(\quad \setminus \quad ', I \setminus I')\}$, where $\quad '$, for any $\quad \in I_P$, denotes restricted to the predicate symbols which are not in Neg_P^*, and $\varrho : I_P \times I_P \to \mathbb{R}$ is defined by

$$
\begin{aligned}
\varrho(J,K) &= \max\{d(J,K), u(J), u(\quad)\} \\
&= \max\{d(J,K), f(\quad '), d(\quad \setminus \quad ', I \setminus I'), f(J'), d(J \setminus J', I \setminus I')\}.
\end{aligned}
$$

We call ϱ the d-metric *associated with* P, and we will show next that it is complete.

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

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

Proposition 3.1.9 yields that ϱ is a complete d-ultrametric on I_P using Lemma 5.3.1.

5.3.2 Proposition Let P be a Φ^*-accessible program with respect to a level mapping l and a model I. Let the d-metric d_3 be defined for P as in equation (5.2) of Definition 5.2.6 for acceptable programs. Then d_3 coincides with ϱ as defined above.

Proof: Clearly, f and ρ coincide, and we obtain $u(\quad) = \max\{\rho(\quad), d_1(\quad I)\}$ for all $\quad \in I_P$. Since $d_2(J,K) \leq d(J,K)$ for all $J,K \in I_P$, it now remains to show that $d(J,K) \leq d_3(J,K)$. So assume that $d(J,K) = 2^{-n}$, where J and $\quad$ differ on some atom $A \in B_P$ or level n which is contained in $\mathtt{Neg}_P^*$. But then either J and I or $\quad$ and I differ on A, hence either $d_1(J, I)$ or $d_1(\quad I)$ is greater than or equal to 2^{-n}. If $A \in \mathtt{Neg}_P^*$, then $d_2(J,K) \geq 2^{-n}$ which suffices. ■

5.3.3 Proposition Let P be a Φ^*-accessible program and ϱ its associated d-metric. If $(\ _n)$ is a sequence which converges in ϱ to some , then $(\ _n)$ converges in the atomic topology on I_P.

Proof: It is easy to see that if $\varrho(\ _n, K) \quad 2^{-k}$, then $\ _n$ and agree on all atoms of level less than k which suffices. ■

The proof of the following proposition carries over from the treatment of acceptable programs in [Fit94], cf. also Proposition 5.4.2.

5.3.4 Proposition Let P be Φ^*-accessible and let ϱ be defined as above. Then the associated immediate consequence operator T_P is a contraction on (I_P, ϱ) with contractivity factor $\frac{1}{2}$.

By the Matthews theorem 1.4.6 we can now conclude the following theorem.

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

Proof: Let P be Φ^*-accessible. Then (I_P, ϱ) is a complete d-ultrametric space and T_P is a contraction relative to ϱ. By Theorem 1.4.6, T_P has a unique fixed point which is the unique supported model of P, and this fixed point can be obtained as the limit, in ϱ, of iterates of T_P. By Proposition 5.3.3, the model can be obtained as stated. ■

We note the following relationship between Φ^*-accessible and acceptable programs. If P is a Φ^*-accessible program, then it is possible to reorder the body literals in each clause from ground(P) such that the resulting ground program is acceptable. Thus Φ^*-accessible programs can be understood as "non-deterministic" acceptable programs. Note, however, that it is not in general possible to reorder the clauses in P itself in order to obtain an acceptable program, which can be seen from the following example.

5.3.6 Program Let P be the program consisting of the following clauses.

$$\begin{aligned}
&p(0) \leftarrow \\
&p(1) \leftarrow r(1) \\
&q(1) \leftarrow \\
&q(0) \leftarrow r(0) \\
&r(x) \leftarrow \neg p(x), \neg q(x)
\end{aligned}$$

This program is not acceptable, nor is the program obtained by swapping the two body atoms in the last clause. However, the program is Φ^*-accessible with respect to the level mapping l with $l(p(0)) = l(q(1)) = 0$, $l(r(0)) = l(r(1)) = 1$ and

$l(p(1)) = l(q(0)) = 2$. Consequently, we are able to obtain a ground acceptable program from ground(P) as

$$
\begin{aligned}
&p(0) \leftarrow \\
&p(1) \leftarrow r(1) \\
&q(1) \leftarrow \\
&q(0) \leftarrow r(0) \\
&r(1) \leftarrow \neg q(1), \neg p(1) \\
&r(0) \leftarrow \neg p(0), \neg q(0).
\end{aligned}
$$

5.4 Φ^*-Accessible Programs

We carry over the results from Section 5.3 to Φ^*-accessible programs.

In the following, P is a Φ^*-accessible program which satisfies the defining conditions with respect to a model I and a level mapping $l : B_P \to \gamma$. We let $\Gamma = \{2^{-\alpha} \mid \alpha \leq \gamma\}$ be ordered as in Section 3.2 and denote $2^{-\gamma}$ by 0.

For $J,K \in I_P$ we define $d(\quad JK,= 0$ and $d(J,K) = 2^{-\alpha}$, where J and differ on some atom $A \in B_P$ of level α, but agree on all ground atoms of lower level, i.e. d coincides with the gum d_l induced by l, see Proposition 5.1.6. As was pointed out in Section 5.1, (I_P, d) is a spherically complete generalized ultrametric space. We also define a function f on I_P by setting $f(\quad) = 0$ if I and $f(\quad) = 2^{-\alpha}$, where α is the smallest integer such that there is an atom $A \in B_P$ with $l(A) = \alpha$, $\models A$ and $I \models A$. Finally, we define a function u on I_P by $u(\quad) = \max\{f(\quad'), d(\quad \setminus \quad', I \setminus I')\}$, where $'$, for any $\in I_P$, is restricted to the predicate symbols which are not in Neg_P^*, and we define a distance function ϱ by

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

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

Proof: (Ui), (Uiii) and (Uiv) follow from Proposition 3.4.4. For spherical completeness let $(\mathcal{B}_\alpha)$ be a chain of nonempty balls in X with midpoints J_α. Let J be the set of all atoms which are *eventually* in J_α, i.e. the set of all $A \in B_P$ such that there exists some β with $A \in J_\alpha$ for all $\alpha \geq \beta$. It is easy to see that for each ball $B_{2^{-\beta}}$ in the chain we have $d(J_\beta, J) \leq 2^{-\beta}$ and hence J is in the intersection of the chain. ■

The proof of the next proposition is analogous to [Fit94, Lemma 7.1 and Proposition 7.1].

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

$\varrho(J,K)$. In particular we have the following, where for any $\in I_P$ we denote by $'$ the set restricted to the predicate symbols which are not in Neg_P^*:

(i) $d(T_P(J) \setminus T_P(J)', I \setminus I')$ $(J \setminus J', I \setminus I')$.

(ii) $f(T_P(\ )')$ (J,K). $<\varrho$

(iii) $d(T_P(J), T_P(\ ))$ (J,K). $<\varrho$

Proof: By symmetry, it suffices to prove properties (i), (ii) and (iii). For convenience, we again identify Neg_P^* with the subset of B_P containing predicate symbols from Neg_P^*.

(i) First note that $d(T_P(J) \setminus T_P(J)', I \setminus I') = d(T_{P^-}(J), I \setminus I')$ since these values only depend on the atoms in Neg_P^*. Let $d(J \setminus J', I \setminus I') = 2^{-\alpha}$. We show that $d(T_{P^-}(J), I \setminus I') \le 2^{-(\alpha+1)}$. So we know that $J \setminus J'$ and $I \setminus I'$ agree on all ground atoms of level less than α and differ on an atom of level α. It suffices to show now that $T_{P^-}(J)$ and $I \setminus I'$ agree on all ground atoms of level less than or equal to α.

Let A be a ground atom in Neg_P^* with $l(A) \le \alpha$ and suppose that $T_{P^-}(J)$ and $I \setminus I'$ differ on A. Assume first that $A \in T_{P^-}(J)$ and $A \in I \setminus I'$. Then there must be a ground instance $A \leftarrow L_1, \dots, L_m$ of P^- such that $J \setminus J' \models L_1, \dots, L_m$. Since $I \setminus I'$ is a fixed point of T_{P^-} and $A \in T_{P^-}(J)$, there must also be a k such that $L_k \in I \setminus I'$, and $l(L_k)$ $l(A)$ by Definition 5.0.4. So we obtain $I \setminus I' \models L_k$ but $J \setminus J' \models L_k$ with $l(L_k)$ which is a contradiction to the assumption that $J \setminus J'$ $<\alpha$ and $I \setminus I'$ agree on all atoms of level less than α. Now assume that $A \in I \setminus I'$ and $A \in T_{P^-}(J)$. It follows that there is a clause $A \leftarrow L_1, \dots, L_m$ in P^- such that $I \setminus I' \models L_1, \dots, L_m$ and $l(A) > l(L_1), \dots, l(L_m)$ by Definition 5.0.4. But then $J \setminus J' \models L_1, \dots, L_m$ since $J \setminus J'$ and $I \setminus I'$ agree on all atoms of level less than α and consequently $A \in T_{P^-}(J)$. This establishes (i).

(ii) Assume $\varrho(J,K) = 2^{-\alpha}$. We show that $f(T_P(\ )') \le 2^{-(\alpha+1)}$, for which in turn we have to show that for each $A \in T_P(\ )$ not in Neg_P^*, i.e. $A \in T_P(\ )'$, with $l(A) \le \alpha$ we have $A \in I'$. Assume that $A \in I'$ for such an A. Since $A \in T_P(\ )'$, there is a ground instance $A \leftarrow L_1, \dots, L_m$ of a clause in P with $\models L_1, \dots, L_m$, and note that A is not in $\mathtt{Neg}_P^*$. Since $A \in I'$, we have $A \in I$ and there must also be a k with $L_k \in I$ and $l(A) > l(L_k)$ by Definition 5.0.4. If L_k belongs to Neg_P^* then, since and I agree on all atoms in Neg_P^* of level less than α, we obtain $\models L_k$ which contradicts $\models L_1, \dots, L_m$. If L_k does not belong to Neg_P^* then it is an atom and since $f(\ ') \le 2^{-\alpha}$, we obtain $I \models L_k$, which is again a contradiction.

(iii) Let $\varrho(J,K) = 2^{-\alpha}$, and let $A \in B_P$ with $l(A) \le \alpha$. It suffices to show that $A \in T_P(\ )$ if and only if $A \in T_P(J)$. We consider two cases.

<u>Case 1</u> $A \in \mathtt{Neg}_P^*$. Since $\varrho(J,K) \le 2^{-\alpha}$, we know that J, and I agree on all atoms in $\mathtt{Neg}_P^*$ of level less than α. Now if $A \in I$, then there is a clause $A \leftarrow L_1, \dots, L_m$ in ground(P^-) with $I \models L_1, \dots, L_m$ and by Definition 5.0.4 we obtain $J \models L_1, \dots, L_m$ and $\models L_1, \dots, L_m$, hence $A \in T_P(\ ) \cap T_P(J)$. If $A \in I$ then for all clauses $A \leftarrow \mathtt{body}$ in ground(P) there is some L in $\mathtt{body}$ with $I \models L$

and $l(L)$ α, and consequently $J \models L$ and $\models L$. We conclude that A is neither in$T_P(J)$ nor in $T_P(\ )$ as required.
<u>Case 2</u> $A \in \mathtt{Neg}_P^*$. Since $\varrho(J,K) \leq 2^{-\alpha}$, we know that J, and I agree on all atoms in $\mathtt{Neg}_P^*$ of level less than α, and that for each $B \in (J \quad)$ not in $\mathtt{Neg}_P^*$ with $l(B)$ we have $B \in I$. Now suppose $A \in I$ with $l(A) \leq \alpha$. Then there is $<\alpha$ a clause $A \leftarrow \mathtt{body}$ in ground(P) with $I \models \mathtt{body}$ and $l(B)$ for all B occurring in $\mathtt{body}$. Consequently, we obtain $J \models \mathtt{body}$ and $\models \mathtt{body}$, so $A \in T_P(J)$ and $A \in T_P(\ )$. Assuming $A \in I$, we know that for each clause $A \leftarrow \mathtt{body}$ in ground(P) there is a literal L in $\mathtt{body}$ such that $I \models L$ and $l(L)$. It suffices to show now that $J \models L$ and $\models L$. Now if L is in $\mathtt{Neg}_P^*$, we obtain $J \models L$ and $\models L$. If L is not in $\mathtt{Neg}_P^*$, then since $I \models L$ we obtain $J \models L$ and $\models L$ which suffices. ■

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

Proof: By Proposition 5.4.2, T_P is strictly contracting with respect to ϱ, which in turn is a spherically complete dislocated generalized ultrametric. By Theorem 1.5.1, the operator T_P must have a unique fixed point which yields a unique supported model for P. ■

By the proof of Theorem 1.5.1 given in Section 3.4, together with the alternative proof of the Prieß-Crampe and Ribenboim theorem in the version of Theorem 1.3.9, we can furthermore obtain the unique model by constructing the sequence $f^\beta(\emptyset)$ as in the proof. It remains to investigate how to obtain $f^\beta(\emptyset)$ in the case that β is a limit ordinal. To this end, we employ the construction from the proof of Proposition 5.4.1, i.e. we set $f^\beta(\emptyset)$ to be the set of all $A \in B_P$ which are eventually in $(f^\alpha(\emptyset))_{\alpha<\beta}$.

5.5 Φ-Accessible Programs

Given a Φ-accessible program P, we define a dislocated generalized ultrametric on I_P which will again allow us to apply the dislocated Prieß-Crampe and Ribenboim theorem, Theorem 1.5.1.

In the following, P is a Φ-accessible program which satisfies the defining conditions with respect to a model I and a level mapping $l : B_P \to \gamma$. As before, we let $\Gamma = \{2^{-\alpha} \mid \alpha \leq \gamma\}$ be ordered as above and denote $2^{-\gamma}$ by 0, and for $J,K \in I_P$ we define the generalized ultrametric d on I_P to be the generalized ultrametric d_l induced by l.

We note that T_P is in general not strictly contracting with respect to d for Φ-accessible programs, even if it is definite.

5.5.1 Program Let P be the following program.

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

For $\quad = \{s^5(0)\}$ and $J = \{s^3(0)\}$ we obtain $d(J,K) = 2^{-3}$. However $d(T_P(\quad), T_P(J) = 2^{-2})$, so T_P is not strictly contracting.

We now define

$$\varrho(J,K) = \max\{d(J,I), d(\quad I)\}K,$$

for all $J,K \in I_P$.

5.5.2 Proposition (X, ϱ) is a spherically complete generalized dislocated ultrametric space.

Proof: If follows from Proposition 3.4.5 that ϱ is a d-gum. Spherical completeness follows from the fact that every nonempty ball contains I. ■

5.5.3 Proposition Let P be Φ-accessible. Then T_P is strictly contracting with respect to ϱ.

Proof: Let $J,K \in I_P$ and assume that $\varrho(J,K) = 2^{-\alpha}$. Then J,K,I agree on all ground atoms of level less than α. We show that $T_P(J)$ and I agree on all ground atoms of level less than or equal to α. A similar argument shows that $T_P(\quad)$ and I agree on all ground atoms of level less than or equal to α, and this suffices.

Let $A \in T_P(J)$ with $l(A) \leq \alpha$. Then there must be a clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) such that $J \models L_1 \qquad L_n$. Since I and J agree on all ground atoms of level less than α, condition (ii) of Definition 5.0.4 cannot hold, because if $I \models L_i$ with $l(A) > l(L_i)$, then $J \models L_i$ and consequently $J \models L_1 \qquad L_n$, which is a contradiction. Therefore, condition (i) of Definition 5.0.4 holds and so $A \in T_P(I) = I$. Hence, $A \in I$.

Conversely, suppose that $A \in I$. Since $I = T_P(I)$, there must be a clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) such that $I \models L_1 \qquad L_n$. Thus, condition (i) of Definition 5.0.4 must hold, and so we can assume that $A \leftarrow L_1, \ldots, L_n$ also satisfies $l(A) > l(L_i)$ for $i = 1, \ldots, n$. Since I and J agree on all ground atoms of level less than α, we have $J \models L_1 \qquad L_n$ and hence $A \in T_P(J)$ as required. ■

5.5.4 Theorem Each Φ-accessible program P has a unique supported model.

Proof: Since P is Φ-accessible, the distance function ϱ as defined above is a spherically complete d-gum. By Proposition 5.5.3, T_P is strictly contracting, hence has a unique fixed point by Theorem 1.5.1. ■

section	class of programs	space	theorem
5.1	acyclic	metric	1.2.2
5.1	locally hierarchical	gum	1.3.4
5.2	acceptable	d-metric	1.4.6
5.3	Φ^*_ω-accessible	d-metric	1.4.6
5.4	Φ^*-accessible	d-gum	1.5.1
5.5	Φ-accessible	d-gum	1.5.1

Table 5.1: Chapter overview: Classes of programs and applied theorems.

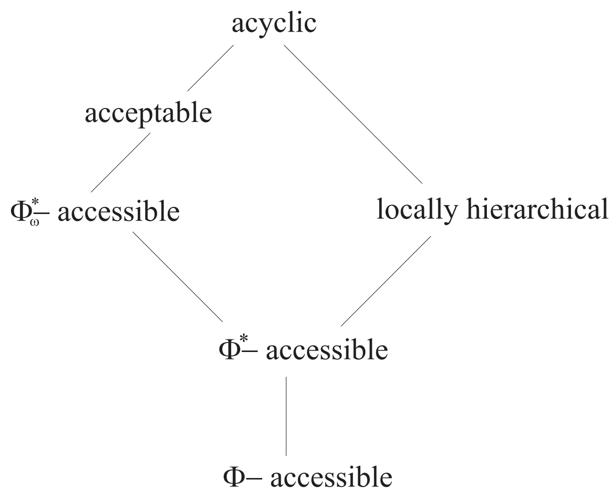

Figure 5.1: Dependencies between classes of programs. If a class is depicted lower in the diagram, this indicates that it is more general.

The proof of Theorem 1.5.1 furthermore yields $\varrho(M, M) = 0$ for the unique fixed point M of T_P. Since the only point of X which has non-zero distance from itself is I, we conclude that $I = M$ is the unique supported model of P. This is somewhat unfortunate since I was needed in order to construct ϱ.

5.6 Summary and Further Work

Chapter 5 can be considered the central chapter in this thesis, with the previous chapters providing applicable results, and the subsequent chapters focussing on a deeper study of the classes of programs and concepts presented in this chapter.

Table 5.1 gives a summary of which fixed-point theorems were applied to which class of programs. Figure 5.1 displays dependencies between the classes described in this chapter. Note that we have not shown yet that every Φ^*-accessible program is Φ-accessible, which we will do in Chapter 6, Theorem 6.5.3.

The fundamental construction used in this chapter is the generalized ultrametric d_l induced by a level mapping l, in the characterization of Proposition

5.1.6. All generalized metric structures employed in this chapter make use of it, and refine it. Investigations remain to be done concerning the possibilities of extending this approach to other semantic operators, probably even operators on many-valued logics as in Section 4.3. Some other questions which arise out of the results in this chapter will be addressed in the rest of this thesis.

Chapter 6

Fitting-style Semantics

In this chapter, we will analyze and characterize unique supported model classes by means of certain three-valued logics. In particular, in Section 6.1 we will introduce three different three-valued logics and their associated consequence operators, and study the relationships between them. In Sections 6.2 and 6.3, we will characterize acceptable and locally hierarchical programs by means of the behaviour of these operators. We will also give alternative constructions of their canonical level mappings. Prompted by the studies of acceptable and locally hierarchical programs, we will define two classes of programs denoted by $[\Phi^*]$ and $[\Phi]$, which will later on turn out to coincide with the classes of all Φ^*-accessible, respectively, Φ-accessible programs. We study these classes in Sections 6.4 and 6.5. Moreover, we will show that the class $[\Phi^*]$ is computationally adequate under SLDNF-resolution.

Many-valued logics have been employed in several studies of the semantics of logic programs. In particular, they have been used to assign special truth values to atoms which possess certain computational behaviour such as being nonterminating [Fit85, Myc84], being ill-typed [Nai98], being floundering [And97], or failing when backtracking [BFMS98]. The motivation for the definitions of the three-valued logics we will be using in the sequel comes from a couple of sources. Primarily, these logics are formulated in order to allow for easy analysis and characterization of the programs or classes of programs in question by using the logic to mimic the defining property of the program or class of programs. This idea is akin to some of those considered in the papers just cited, and is a component of work presented in Section 5.2 where a program transformation which outputs a locally hierarchical program, when input an acceptable one, is used in the characterization of acceptable programs. Natural questions, partly answered here, then arise as to the different ways that different classes of programs can be characterized. On the other hand, some of the work in this chapter can also be viewed as a contribution to the asymmetric semantics proposed in [FBJ90] where it is noted that certain differences between Pascal, LISP and Prolog, for example, are easily described in terms of three-valued logic. Thus, [FBJ90] is also a source of motivation for our definitions. However, we note that all programs analyzed in this chapter do have unique supported models, therefore the third truth value *undefined* will only be used for obtaining the unique supported two-valued model. Hence, interpretations of *undefined* from the point of view of computation (such as non-halting) are not actually necessary in this chapter.

All semantical considerations presented in this paper are with respect to arbitrary preinterpretations.

6.1 Three-valued Logics

A *three-valued interpretation* of a program P is a pair (T, F) of disjoint sets $T, F \quad B_P$. Note that the notation used here is different from the one of Section 4.3, but is easily seen to be equivalent. Given such an interpretation $I = (T, F)$, a ground atom A is true (t) in I if $A \in T$, false (f) in I if $A \in F$, and undefined (u) otherwise; $\neg A$ is true in I if and only if A is false in I, $\neg A$ is false in I if and only if A is true in I and $\neg A$ is undefined in I if and only if A is undefined in I.

Given $I = (T, F)$, we denote T by I^+ and F by I^-. Thus, $I = (I^+, I^-)$. If $I^+ \quad I^- = B_P$, we call I a *total* three-valued interpretation of the program P. Total three-valued interpretations can be identified with elements of I_P.

Given a program P, the set $I_{P,3}$ of all three-valued interpretations of P forms a complete partial order (in fact, complete semi-lattice) with the ordering $\leq$ defined by

$$I \leq \qquad \text{if and only if} \qquad I^+ \qquad {}^+ \text{ and } I^- \qquad {}^-$$

with least element $(\emptyset, \emptyset)$ which we will denote by $\perp$. Notice that total three-valued interpretations are maximal elements in this ordering.

In our present context, it will be sufficient to give truth tables for conjunction and disjunction, and we will make use of three different three-valued logics which we are now going to define. It should be noted here that the truth tables for disjunction are the same in all three logics and that disjunction is commutative.

The first logic, which we will denote by $\mathcal{L}_1$, evaluates conjunction as in Fitting's Kripke-Kleene semantics [Fit85] (in fact, as in Kleene's strong three-valued logic, see [FBJ90]). This work built on [Myc84] and was subsequently studied in the literature e.g. in [Kun87, AP93, Nai98]. Disjunction will be evaluated differently though, as indicated by the truth table in Table 6.1.

The second three-valued logic, $\mathcal{L}_2$, will be used for studying acceptable programs and is non-commutative under conjunction. It will be sufficient to evaluate u   f to u instead of f and leaving the truth table for $\mathcal{L}_1$ otherwise unchanged. This way of defining conjunction was employed in [And97] and [BFMS98], see also the discussion of LISP in [FBJ90]. The truth table is again given in Table 6.1.

The third logic, $\mathcal{L}_3$, will be used for studying locally hierarchical and acyclic programs. For this purpose, we use a commutative version of $\mathcal{L}_2$ where we evaluate f   u to u instead of f, see the discussion in [FBJ90] of Kleene's weak three-valued logic in relation to Pascal. The truth table is shown in Table 6.1.

Let P be a normal logic program, and let $\mathcal{L}_i$ denote one of the three-valued logics above, where $i = 1, 2$ or 3. Corresponding to each of these logics we define

		Logic $\mathcal{L}_1$		Logic $\mathcal{L}_2$		Logic $\mathcal{L}_3$	
p	q	p q	$p \vee q$	p q	$p \vee q$	p q	$p \vee q$
t	t	t	t	t	t	t	t
t	u	u	u	u	u	u	u
t	f	f	t	f	t	f	t
u	t	u	u	u	u	u	u
u	u	u	u	u	u	u	u
u	f	f	u	u	u	u	u
f	t	f	t	f	t	f	t
f	u	f	u	f	u	u	u
f	f	f	f	f	f	f	f
Operator		$\Phi_{P,1} = \Phi_P$		$\Phi_{P,2}$		$\Phi_{P,3}$	

Table 6.1: Truth tables for the logics $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$.

an operator F_P on $I_{P,3}$ as follows. For $I \in I_{P,3}$, let $F_P(I) = (T, F)$ where T denotes the set

$$\{A \in B_P \mid \text{there is } A \leftarrow \texttt{body} \in \text{ground}(P) \text{ s.t. } \texttt{body} \text{ is true}_i \text{ in } I\},$$

and F denotes the set

$$\{A \in B_P \mid \text{for every } A \leftarrow \texttt{body} \in \text{ground}(P), \texttt{body} \text{ is false}_i \text{ in } I\}.$$

Of course, true$_i$ and false$_i$ here denote truth respectively falsehood in the logic $\mathcal{L}_i$. Notice that if A is not the head of any clause in P, then A is false in $F_P(I)$ for any I.

It is clear that F_P is monotonic in all three cases. We set $F_P{\uparrow}0 = \bot$,

$$F_P{\uparrow}\alpha = F_P(F_P{\uparrow}(\alpha - 1)) \text{ for } \alpha \text{ a successor ordinal, and}$$
$$F_P{\uparrow}\alpha = \bigcup_{\beta<\alpha} F_P{\uparrow}\beta \text{ for } \alpha \text{ a limit ordinal.}$$

Since F_P is monotonic, it has a least fixed point by the Knaster-Tarski theorem 1.1.7 which is equal to $F_P{\uparrow}\alpha$ for some ordinal α called the *closure ordinal* of P (for the chosen logic $\mathcal{L}_i$).

Throughout the sequel, we will denote F_P by $\Phi_{P,1}$, $\Phi_{P,2}$ or $\Phi_{P,3}$ if the chosen logic is correspondingly $\mathcal{L}_1$, $\mathcal{L}_2$ or $\mathcal{L}_3$. The appropriate symbol is also included in Table 6.1 for ease of reference. Note that the behaviour of each of these operators depends only on the evaluation of conjunction. In fact, $\Phi_{P,1}$ is the very same operator as used in [Fit85]. We will also denote this operator by Φ_P.

6.1.1 Proposition If we evaluate implication such that the partial truth table in Table 6.2 is satisfied, then for each $i = 1, 2, 3$, $\Phi_{P,i}$ is a local consequence operator.

p	q	$p \leftarrow q$
t	t	t
t	u	t
t	f	t
u	u	t
u	f	t
f	f	t

Table 6.2: Desired implication properties for 3-valued logics.

Proof: Immediate by Definitions 4.3.7 and 4.3.10. ■

6.1.2 Proposition Let P be a normal logic program and let $I, I', I'' \in I_{P,3}$ be such that $I \leq I' \leq I''$. Then we have

$$\Phi_{P,3}(I) \leq \Phi_{P,2}(I') \leq \Phi_{P,1}(I'').$$

Proof: The following observations are clear from the given truth tables, and indeed suffice. If a body of a clause is true (false) in $\mathcal{L}_3$, then it is true (false) in $\mathcal{L}_2$. If it is true (false) in $\mathcal{L}_2$, then it is true (false) in $\mathcal{L}_1$. ■

We investigate the relationship between Φ_P and T_P for a given program P, extending some results in [AP93].

6.1.3 Lemma Let P be a normal logic program, let $I \in I_P$ and let   be a partial interpretation for P with $\ ^{+}\ \ I\ \ ^{c}\ ^{-}$. Then $\Phi_P(\ )^{+}\ \ T_P(I)\ \ ^{c}\Phi_P(\ )^{-}$. Furthermore, if $\ ^{+} = I = ^{c}\ ^{-}$, so that   is total, then $\Phi_P(\ )^{+} = T_P(I) = {}^{c}\Phi_P(\ )^{-}$.

Proof: Let $A \in \Phi_P(\ )^{+}$. Then A must be the head of a clause $A \leftarrow A_1, \dots, A_{k_1}, \neg B_1, \dots, \neg B_{k_2}$ in ground(P) with $A_i \in\ ^{+}$ and $B_j \in\ ^{-}$ for all $i = 1, \dots, k_1$ and $j = 1, \dots, k_2$. By assumption, it follows that for these values of i and j, $A_i \in I$ and $B_j \in I$, and hence $A \in T_P(I)$.
For the second inclusion, it suffices to show that $\Phi_P(\ )^{-}\ \ ^{c}T_P(I)$. Let $A \in \Phi_P(\ )^{-}$. Then, for every clause $A \leftarrow A_1, \dots, A_{k_1}, \neg B_1, \dots, \neg B_{k_2}$ in ground(P), we have some $A_i \in\ ^{-}$ or some $B_j \in\ ^{+}$. Hence, for every such clause, we have some $A_i \in I$ or some $B_j \in I$, which implies that $A \in T_P(I)$.
For the last statement, it suffices to note that a conjunction $L_1\ \ \ L_n$ of literals is true in   if and only if it is true in I if and only if it is not false in . ■

The following straightforward corollary provides the essential link between the Φ-operator, the single-step operator T_P and convergence in Q. Intuitively speaking, iterates of T_P are "squeezed between" the iterates of Φ_P.

6.1.4 Corollary Let $I_n = T_P^n(\emptyset)$ and let $\ _n = \Phi_P \uparrow n$. Then, for all $n \in \mathbb{N}$, we obtain $\ _n^{+}\ \ I_n\ \ ^{c}\ _n^{-}$.

The following now easily carries over from [AP93], and is in fact a direct consequence of Lemma 6.1.3.

6.1.5 Proposition Let P be a normal logic program and let $I = (I^+, I^-)$ be a total interpretation for P. Then I is a fixed point of Φ_P if and only if I^+ is a fixed point of T_P. Furthermore, if Φ_P has exactly one total fixed point M, then M^+ is the unique fixed point of T_P.

Proof: Let I be a fixed point of Φ_P. Then $I^+ \quad I^+ \quad {}^cI^-$ and by Lemma 6.1.3 we obtain $I^+ = \Phi_P(I)^+ \quad T_P(I^+) \quad {}^c\Phi_P(I)^- = {}^cI^- = I^+$. Conversely, let I^+ be a fixed point of T_P. By Lemma 6.1.3, we obtain $\Phi_P(I)^+ = T_P(I^+) = I^+ = {}^cI^- = {}^c\Phi_P(I)^-$, and therefore $\Phi_P(I)^+ = I^+$ and $\Phi_P(I)^- = I^-$. The last statement now follows immediately. ■

Collecting together the previous results now yields convergence in Q of iterates of T_P.

6.1.6 Proposition Let P be a normal logic program and assume that $M = \Phi_P\!\uparrow\omega$ is total. Then $T_P^n(\emptyset)$ converges in Q to M^+, and M^+ is the unique supported model M_P of P.

Proof: Using the notation from Corollary 6.1.4, we obtain $M^+ = \quad {}^+_n$ and $M^- = \quad {}^-_n$. Since M is total, we obtain from Propositions 4.2.2 and 6.1.5 that M^+ is the limit in Q of the sequence I_n. Since totality of $\Phi_P\!\uparrow\omega$ implies that it is the unique fixed point of Φ_P, it therefore equals (M^+, M^-), so that M^+ is the unique fixed point of T_P by Proposition 6.1.5. ■

6.1.7 Proposition Let P be a normal logic program, let F_P denote $\Phi_{P,i}$, for $i = 1, 2, 3$, and assume that $M = F_P\!\uparrow\alpha$ is total, where α is the corresponding closure ordinal of P. Then M^+ is the unique two-valued supported model of P. Furthermore, the transfinite sequence $(F_P\!\uparrow\beta)_\beta$ converges in the atomic topology to M^+.

Proof: By totality of M, Propositions 6.1.2 and 6.1.5 we obtain M^+ as a fixed point of T_P. Since M is the least fixed point of F_P and is maximal in $I_{P,3}$, it is the unique fixed point of F_P. The convergence results follows as in Proposition 6.1.6. ■

Given a ground atom A which occurs as the head of an element $A \leftarrow C$ of ground(P), we form the *pseudo clause*, or simply *clause*, $A \leftarrow \bigvee_i C_i$ whose *body* $\bigvee_i C_i$ is the (possibly infinite) disjunction of the bodies C_i of all clauses in ground(P) whose head is A; we call A the *head* of the pseudo clause $A \leftarrow \bigvee_i C_i$. The set of all such pseudo clauses will be denoted by P^*. It will be convenient to assign "truth" values to $\bigvee_i C_i$, relative to the logics $\mathcal{L}_i$ by in fact assigning truth values to arbitrary disjunctions of literals and then employing the same sort of abuse for "disjunctions" of ground literals which was established earlier for conjunction. This is done as follows: ${}_{i}C_i$ will be assigned value *true* (t) if and only if at least one C_i is true and none are undefined; it will be assigned value *undefined* (u) if and only if at least one C_i is undefined; it will be assigned value *false* (f) if and only if all the C_i are false. These definitions are the natural extension to possibly infinite disjunctions of the values given iteratively to finite disjunctions by the truth tables in Table 6.1.

Letting F_P denote any one of the $\Phi_{P,i}$, for $i = 1, 2, 3$, we define an operator F_{P^*} on $I_{P,3}$ as follows. For $I \in I_{P,3}$, set $F_{P^*}(I) = (T, F)$, where T is the set of all ground atoms which occur as the head of a pseudo clause in P^* whose body is true in I, and F is the set of all ground atoms which occur as the head of a pseudo clause whose body is false in I. As before, $\Phi_{P^*,i}$ will denote F_{P^*} when the chosen logic is $\mathcal{L}_i$, $i = 1, 2, 3$. Note that F_{P^*} is again monotonic for any choice of underlying logic. Ordinal powers $F_{P^*} \uparrow \alpha$ are defined as for F_P. We will denote the operator $\Phi_{P^*,i}$ also by $\Phi^*_{P,i}$, and $\Phi^*_{P,1}$ by Φ^*_P.

6.1.8 Example We give an example illustrating the program transformation P^*. Let P be the (propositional) program

$$
\begin{aligned}
&a \leftarrow b\\
&a \leftarrow c\\
&b \leftarrow\\
&c \leftarrow c
\end{aligned}
$$

then P^* is

$$
\begin{aligned}
&a \leftarrow b \vee c\\
&b \leftarrow\\
&c \leftarrow c
\end{aligned}
$$

Let I be the three-valued interpretation $(\{b\}, \emptyset)$. Then $\Phi_{P,1}(I) = (\{a, b\}, \emptyset)$, which is also the least fixed point of $\Phi_{P,1}$. However, since c is undefined in I, we have $\Phi_{P^*,1}(I) = (\{b\}, \emptyset)$, which is the least fixed point of $\Phi_{P^*,1}$. The difference between $\Phi_{P,1}$ and $\Phi_{P^*,1}$ results from the way in which disjunction is defined, see the following proposition, Proposition 6.1.10. In fact, in this context it is worth noting an observation made by one of the referees of [HS99a], as follows. In classical two-valued logic, the programs $(a \leftarrow b)\quad(a \leftarrow c)$ and $a \leftarrow (b \vee c)$ are equivalent simply because of the distributive laws and De Morgan's law that $\neg b \wedge \neg\, c$ and $\neg(b \vee c)$ are equivalent. In the Logics $\mathcal{L}_i$, $i = 1, 2, 3$, $\neg b \wedge \neg\, c$ and $\neg(b \vee c)$ are not equivalent as can easily be verified by, for example, taking b to be true and c to be undefined. In fact, the rule $a \leftarrow (b \vee c)$ with disjunctive body is weaker (leaves more undefined) than the two separate rules $a \leftarrow b$ and $a \leftarrow c$.

6.1.9 Proposition If we evaluate implication such that the partial truth table in Table 6.2 is satisfied, then for each $i = 1, 2, 3$, $\Phi^*_{P,i}$ is a local consequence operator.

Proof: Immediate by Definitions 4.3.7 and 4.3.10. ■

6.1.10 Proposition Let P be a normal logic program and let $I, I', I'' \in I_{P,3}$ be such that $I \leq I' \leq I''$. Then we have

$$\Phi_{P^*,3}(I) \leq \Phi_{P^*,2}(I') \leq \Phi_{P^*,1}(I''),$$

and for F denoting any of the Φ_i, for $i = 1, 2, 3$, we have

$$F_{P^*}(I) \leq F_P(I) \qquad \text{and} \qquad F_{P^*}(I)^- = F_P(I)^-.$$

Proof: The proof is along the same lines as the proof of Proposition 6.1.2 noting that in a disjunction $\;_i C_i$ which is true, no C_i is undefined. ■

6.2 Acceptable Programs

We are able to characterize acceptable programs by means of the operator $\Phi_{P^*,2}$, and we do this next. We will need the following proposition.

6.2.1 Proposition Suppose that P is acceptable with respect to a level mapping l. Then $M_P = \Phi_{P,1}{\uparrow}\omega$ is total, M_P^+ is the unique supported model of P and P is acceptable with respect to l and M_P^+.

Proof: The first statement carries over directly from [AP93], where it was shown for Herbrand preinterpretations. The second statement was shown in Theorem 5.2.12. ■

6.2.2 Lemma Let P be acceptable. Then $M = \Phi_{P^*,2}{\uparrow}\omega$ is total. Furthermore, $M = \Phi_{P,2}{\uparrow}\omega$, and M^+ is the unique supported model M_P^+ of P.

Proof: Let l be a level mapping with respect to which P is acceptable. By Proposition 6.2.1, P is acceptable with respect to l and M_P^+. Assume that there is a ground atom A which is undefined in M. Without loss of generality we can assume that $l(A)$ is minimal. Then by definition of $\mathcal{L}_2$, there is precisely one pseudo clause in P^* of the form $A \leftarrow \;_i C_i$ in which at least one of the C_i, say C_1, is undefined. Thus, there must occur a left-most ground body literal B in C_1 which is undefined in M, and this ground literal is to the left in C_1 of the first ground literal which is false in M. Hence, all ground literals occurring to the left of B must be true in M. Since $M \leq M_P$ by Proposition 6.1.10, all these ground literals must also be true in M_P^+. By acceptability of P we therefore conclude that $l(B) \quad l(A)$, contradicting the minimality of $l(A)$. By Proposition 6.1.10, the second statement holds. The last statement follows from Proposition 6.1.7. ■

6.2.3 Definition Let P be acceptable. Define the mapping l_P as follows: $l_P(A)$ is the lowest ordinal α such that A is not undefined in $\Phi_{P^*,2}\!\uparrow\!(\alpha \quad 1)$.

6.2.4 Proposition Let P be acceptable. Then l_P is an ω-level mapping and P is acceptable with respect to l_P and M_P. Furthermore, if l is another level mapping with respect to which P is acceptable, then $l_P(A) \leq l(A)$ for all $A \in B_P$. In particular, l_P is exactly the canonical acceptable-level mapping defined in Construction 5.2.16.

Proof: By Lemma 6.2.2, l_P is indeed an ω-level mapping.

Let A be the head of a ground clause $\mathcal{C}$ in P with $l_P(A) = n$. Then the body $_i C_i$ of the corresponding pseudo clause in P^* is either true or false (i.e. is not undefined) in $\mathcal{N} = \Phi_{P^*,2}\!\uparrow\! n$. If $_i C_i$ is true, each C_i evaluates to true or false in $\mathcal{N}$. If C_i evaluates to true in $\mathcal{N}$ (and at least one must), then all ground literals in C_i are true in $\mathcal{N}$, and therefore have level less than or equal to $n-1$. If C_i evaluates to false in $\mathcal{N}$, then there must be a ground literal in C_i which is false in $\mathcal{N}$ such that all ground literals occurring to the left of it are true in $\mathcal{N}$. Moreover all these ground literals are not undefined in $\mathcal{N}$ and hence have level less than or equal to $n-1$. A similar argument applies if $_i C_i$ is false in $\mathcal{N}$. Since $\mathcal{N} \leq M_P$, it is now clear that the clause $\mathcal{C}$ satisfies condition (5.1) of acceptability given in Definition 5.0.2 with respect to l_P and M_P.

Now let l be another level mapping with respect to which P is acceptable. By Proposition 6.2.1, P is acceptable with respect to l and M_P. Let $A \in B_P$ with $l(A) = n$. We show by induction on n that $l(A) \geq l_P(A)$. If $n = 0$, then A appears only as the head of unit clauses, and therefore $l_P(A) = 0$. Now let $n > 0$. Then in every clause with head A, the left prefix of the corresponding body, up to and including the first ground literal which is false in M_P, contains only ground literals L with $l(L) \quad n$. By the induction hypothesis, $l_P(L) \quad n$ for all these ground literals L and, consequently, $l_P(A) \leq l(A)$ by definition of l_P.

The last statement follows from Theorem 5.2.21, where it is shown that the given minimality property characterizes l_P. ■

We are now in a position to characterize acceptable programs.

6.2.5 Theorem Let P be a normal logic program. Then P is acceptable if and only if $M = \Phi_{P^*,2}\!\uparrow\!\omega$ is total.

Proof: By Lemma 6.2.2 it remains to show that totality of M implies acceptability. Define the ω-level mapping l_P for P as in Definition 6.2.3. Since M is total, l_P is indeed an ω-level mapping for P. We will show that P is acceptable with respect to l_P and M.

Arguing as in the proof of the previous proposition, let A be the head of a ground clause $\mathcal{C}$ in P with $l_P(A) = n$. Then the corresponding body C evaluates to true or false in $\mathcal{N} = \Phi_{P^*,2}\!\uparrow\! n$. If it evaluates to true in $\mathcal{N}$, then all ground literals in C are true in $\mathcal{N}$, and therefore have level less than or equal to $n-1$. If it evaluates to false in $\mathcal{N}$, then there must be a ground literal in C which is false in $\mathcal{N}$ such that all ground literals occurring to the left of it are true in $\mathcal{N}$. Again, all these ground literals are not undefined in $\mathcal{N}$ and hence have level less than or equal to $n-1$. Since $\mathcal{N} \leq \mathcal{M}$, the clause $\mathcal{C}$ satisfies the condition of acceptability given in Definition 5.0.2. ■

In [Mar96], it was shown that the class of programs which terminate under Chan's constructive negation [Cha88] coincides with the class of programs which are acceptable with respect to a model based on a preinterpretation whose domain is the Herbrand universe and contains infinitely many constant and function symbols, cf. Section 5.2. We therefore obtain the following result.

6.2.6 Theorem A normal logic program P terminates under Chan's constructive negation if and only if $\Phi_{P^*,2}\!\uparrow\!\omega$ is total, where $\Phi_{P^*,2}$ is computed with respect to a preinterpretation whose domain is the Herbrand universe and contains infinitely many constant and function symbols.

We are also able to characterize acceptability as follows.

6.2.7 Proposition A normal logic program P is acceptable if and only if there exists an ω-level mapping l for P and a model I for P such that the following is satisfied: Condition (5.1) of Definition 5.0.2 holds and whenever $I \models \texttt{body}$ for all clauses $A \leftarrow \texttt{body}$ in ground(P), we have $I \models A$.

Proof: Let P be a program which is acceptable with respect to a level mapping l and a model I. Then P is acceptable with respect to its unique supported model M and l by Theorem 5.2.12, so condition (5.1) is satisfied with respect to M. Since M is supported, the additional condition is also satisfied with respect to M.

Conversely, let l and I be such that condition (5.1) and the additional condition in the statement of the proposition are satisfied. Since I is a model and the additional condition holds, we obtain that I is a supported model. So I, restricted to the predicate symbols in $\texttt{Neg}^*_P$, is a supported model of P^- which suffices. ■

6.3 Locally Hierarchical Programs

We will now give a new characterization of locally hierarchical and acyclic programs along the lines of Theorem 6.2.5, using the operator $\Phi_{P^*,3}$.

6.3.1 Lemma Let P be locally hierarchical with respect to the level mapping l and let $A \in B_P$ be such that $l(A) = \alpha$. Then A is true or false in $\Phi_{P^*,3}\!\uparrow\!(\alpha \quad 1)$. In particular, there exists an ordinal α_P such that $\Phi_{P^*,3}\!\uparrow\!\alpha_P$ is total.

Proof: The proof is by transfinite induction on α. The base case follows directly from the fact that if $\alpha = 0$, then A appears as head of unit clauses only. Now let $\alpha = \beta \quad 1$ be a successor ordinal. Then all ground literals appearing in bodies of clauses with head A have level less than or equal to β. By the induction hypothesis, they are all not undefined in $\Phi_{P^*,3}\uparrow(\beta \quad 1)$ and therefore A is either true or false in $\Phi_{P^*,3}\uparrow(\alpha \quad 1)$. If α is a limit ordinal, then all ground literals occurring in bodies of clauses with head A have level strictly less than α. Hence, by the induction hypothesis and since α is a limit ordinal, all these ground body literals are not undefined in $\Phi_{P^*,3}\uparrow\alpha$, and therefore A is true or false in $\Phi_{P^*,3}\uparrow(\alpha \quad 1)$. ■

6.3.2 Corollary Let P be a locally hierarchical program with level mapping $l : B_P \to \alpha$ and let $M = \Phi_{P,1}\uparrow\alpha$. Then M is total and $M_P = M^+$ is the unique supported model of P.

Proof: By Propositions 6.1.2 and 6.1.10, we have $\Phi_{P^*,3}\uparrow\beta \leq \Phi_{P,3}\uparrow\beta \leq \Phi_{P,1}\uparrow\beta$ for all ordinals β. Since $\Phi_{P^*,3}\uparrow\alpha$ is total by Lemma 6.3.1, the given statement holds using Proposition 6.1.7. ■

6.3.3 Definition Let P be locally hierarchical. Define the level mapping l_P for P as a function $l_P : B_P \to \alpha_P$ where $l_P(A)$ is the least ordinal α such that A is true or false in $\Phi_{P^*,3}\uparrow(\alpha \quad)$. $+1$

6.3.4 Proposition Let P be locally hierarchical with respect to some level mapping l. Then l_P is a level mapping for P and, for all $A \in B_P$, we have $l_P(A) \leq l(A)$. Furthermore, l_P coincides with the canonical lh-level mapping of Construction 5.1.1.

Proof: The mapping l_P is indeed a level mapping by Lemma 6.3.1. Let $A \in B_P$ with $l(A) = \alpha$. We show the given minimality statement by transfinite induction on α. If $\alpha = 0$, then A appears as the head of unit clauses only, and so $l_P(A) = 0$. If $\alpha = \beta \quad 1$ is a successor ordinal, then all ground literals L occurring in bodies of clauses with head A have level $l(L) \leq \beta$. By the induction hypothesis, we obtain $l_P(L) \leq \beta$ for all those ground literals, and so $l_P(A) \leq \alpha = l(A)$ by construction of l_P. If α is a limit ordinal, then all ground literals L occurring in bodies of clauses with head A have level $l(L) \quad$. Since $l_P(L) \leq l(L)$ and since α is a limit ordinal, we obtain that all these ground literals L are not undefined in $\Phi_{P^*,3}\uparrow\alpha$ and therefore $l_P(A) \leq \alpha = l(A)$ as desired.

The last statement follows since the minimality property just proved characterizes the canonical lh-level mapping as was shown in Proposition 5.1.4. ■

Note that it is an easy corollary of the previous results that if a program P is acyclic, then $\Phi_{P^*,3}\uparrow\omega$ is total.

6.3.5 Theorem A normal logic program P is locally hierarchical if and only if $\Phi_{P^*,3}\uparrow\alpha$ is total for some ordinal α. It is acyclic if and only if $\Phi_{P^*,3}\uparrow\omega$ is total.

Proof: Let P be a normal logic program such that $\Phi_{P^*,3}\uparrow\alpha$ is total for some α. We define a mapping $l_P : B_P \to \alpha$ as in Definition 6.3.3. From the definition of the logic $\mathcal{L}_3$ it is now obvious that P is indeed locally hierarchical with canonical lh-level mapping l_P. The reverse was shown in Lemma 6.3.1. The statement for acyclic programs now follows similarly. ■

6.4 Φ^*-Accessible Programs

Our investigations of acceptable and locally hierarchical programs suggest we define a class of programs by the property that $\Phi_{P^*,1}\!\uparrow\!\alpha$ is total for some ordinal α. We will do this next, show that this class contains exactly the Φ^*-accessible programs, and also that this class is computationally adequate.

6.4.1 Definition We define the class $[\Phi^*]$ of normal logic programs as follows. A normal logic program P is contained in $[\Phi^*]$, if $\Phi_{P^*,1}\!\uparrow\!\alpha$ is total for some ordinal α.

6.4.2 Theorem Every program in $[\Phi^*]$ has a unique supported model. Furthermore, this class contains all acceptable and all locally hierarchical programs.

Proof: Immediate by Propositions 6.1.7 and 6.1.10. ■

6.4.3 Definition The *canonical level mapping wrt.* Φ^* for a given program in $[\Phi^*]$ is denoted by l^* and defined as follows. For every $A \in B_P$, set $l^*(A) = \alpha$, where α is the minimal ordinal such that A is true or false in $\Phi_{P^*,1}\!\uparrow\!(\alpha \quad 1)$.

The following is immediate by Proposition 6.1.10.

6.4.4 Proposition If P is acceptable or locally hierarchical with canonical acceptable-level mapping, respectively canonical lh-level mapping, l_P, then $l^*(A) \geq l_P(A)$ for all ground atoms A.

6.4.5 Proposition Let P be a normal logic program. Then P is contained in $[\Phi^*]$ if and only if the following property holds for some model I and some level mapping l for P: For each clause $A \leftarrow L_1, \dots, L_n$ in ground(P), we either have $I \models L_1 \qquad L_n$ and $l(A) > l(L_i)$ for all $i = 1, \dots n$, or there exists $i \in \{1, \dots, n\}$ such that $I \models L_i$, $I \models A$ and $l(A) > l(L_i)$. Furthermore $l^*(A) \leq l(A)$ for every $A \in B_P$.

Proof: The first statement follows immediately from the definition of the logical connectives in the logic $\mathcal{L}_1$, using a proof by transfinite induction.

The minimality property of l^* is shown by transfinite induction along the same lines as in the proofs of the Propositions 6.2.4 and 6.3.4. ■

6.4.6 Corollary $[\Phi^*]$ contains exactly all Φ^*-accessible programs.

Proof: The proof is analogous to the proof of Proposition 6.2.7, using Proposition 6.4.5. ■

It was shown in Section 5.1 that the class of all locally hierarchical programs is computationally adequate in the sense that every partial recursive function can be computed with such a program if the use of safe cuts is allowed. For Φ^*-accessible programs, the cut need not be used, and we will show this next. The proof basically shows that given a partial recursive function, there is a definite program as given in [Llo88] which computes that function. This program will turn out to be a Φ^*-accessible program.

6.4.7 Theorem Let f be a partial recursive function. Then there exists a definite Φ^*-accessible program which computes f.

Proof: We will make use of the definite program P_f given in [Llo88, Theorem 9.6], and we refer the reader to the proof of this theorem for details. It is easily seen that we have to consider the minimalization case only. In [Llo88], the following program P_f was given as an implementation of a function f which is the result of applying the minimalization operator to a partial recursive function g, which is in turn implemented by a predicate p_g. We abbreviate $X_1, \dots, X_n$ by X.

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

This program is not Φ^*-accessible. However, we can replace it with a program P'_f which has the same procedural behaviour and is Φ^*-accessible. In fact, we replace the definition of r by

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

where the predicate lt is in turn defined as

$$
\begin{aligned}
lt(0, s(X)) &\leftarrow \\
lt(s(X), s(Y)) &\leftarrow lt(X, Y)
\end{aligned}
$$

and is obviously Φ^*-accessible. By a straightforward analysis of the original program P_f, it is clear that the addition of $lt(y, z)$ in the second defining clause of r does not alter the procedural behaviour of the program. Since lt and p_g are Φ^*-accessible, it is now easy to see that r is Φ^*-accessible, and so therefore is P'_f. ■

It is worth noting that negation is not needed here in order to obtain full computational power, so Theorem 6.4.7 strenghtens the result of [Llo88] referred to in the proof. By contrast, as already noted, definite locally hierarchical programs seem not to provide full computational power. Regardless of some known drawbacks in SLDNF-resolution, it is interesting to know that relative to it the class of all Φ^*-accessible programs has full computational power neither the class of acyclic nor even the class of acceptable programs has this property.

6.5 Φ-Accessible Programs

We carry over our methods to the study of Φ-accessible programs.

6.5.1 Definition Let P be a normal logic program. Then P is contained in $[\Phi]$ if and only if $\Phi_P \uparrow \alpha$ is total for some ordinal α.

6.5.2 Definition Let P be in $[\Phi]$. For each $A \in B_P$, let $l_P(A)$ denote the least ordinal α such that A is not undefined in $\Phi_P \uparrow (\alpha \quad 1)$. We call the resulting mapping l_P the *canonical level mapping for P wrt. Φ*.

6.5.3 Theorem The class $[\Phi]$ contains exactly the Φ-accessible programs.

Proof: Let P be in $[\Phi]$, let l_P be its canonical level mapping wrt. Φ, let α be its closure ordinal wrt. Φ_P and let $M_P = \Phi_P \uparrow \alpha^+$ be its unique supported (two-valued) model.
(a) Let $A \in M_P$ and $l_P(A) = \beta$. By definition of l_P and Φ_P there exists a clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) such that the $L_1, \ldots, L_n$ are true in $\Phi \uparrow \beta$, and hence are also true in M_P. Again by definition of l_P we obtain $l_P(A) > l_P(L_i)$ for all i.
(b) Let $A \in M_P$ and $l_P(A) = \beta$. By definition of l_P and Φ_P we obtain that for any clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) we must have that $L_1 \qquad L_n$ is false in $\Phi_P \uparrow \beta$. So there must be some i such that L_i is false in $\Phi_P \uparrow \beta$ and $l(L_i) \qquad$ by definition of l_P, and hence $l_P(A) > l_P(L_i)$. Thus, P is Φ-accessible with respect to M_P and l_P.

Conversely, let P be Φ-accessible, so that P satisfies conditions (i) and (ii) of Definition 5.0.4 with respect to a model I and a level mapping l. We show by induction on β that any $A \in B_P$ with $l(A) = \beta$ is not undefined in $\Phi_P \uparrow (\beta \qquad)$ and, furthermore, that I and $\Phi_P \uparrow (\beta \quad 1)$ agree on A.
If $l(A) = 0$, then A must be the head of a unit clause or does not appear in any head. In the first case, A is true in $\Phi_P \uparrow 1$, and in the second case, A is false in $\Phi_P \uparrow 1$. Note that in the first case A is also true in I since condition (i) of Definition 5.0.4 applies and I is a model of P. Also, in the second case, A is also false in I since condition (ii) of Definition 5.0.4 applies.
Now let $l(A) = \beta$. If there is no clause in ground(P) with head A, then A is false in $\Phi_P \uparrow 1 \leq \Phi_P \uparrow (\beta \quad 1)$ and also false in I since condition (ii) of Definition 5.0.4 applies. So assume there is a clause in ground(P) with head A. By definition of Φ-accessibility, either condition (i) or condition (ii) of Definition 5.0.4 applies.
If condition (i) applies, then there is a clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) such that $l(L_1), \ldots, l(L_n) \qquad (A)$ and therefore, by the induction hypothesis, the $<l$
$L_1, \ldots, L_n$ are not undefined in $\Phi_P \uparrow \beta$ and I agrees with $\Phi_P \uparrow \beta$ on them. Now,

since I is a model of P and $I \models L_1, \ldots, L_n$, we obtain that A is true in I and by definition of Φ_P also in $\Phi_P \uparrow \beta$.
If condition (ii) applies, then for each clause $A \leftarrow L_1, \ldots, L_n$ in ground(P) there is some i such that $l(A) > l(L_i)$ and L_i is false in I. Hence we obtain that L_i is false in $\Phi_P \uparrow \beta$ by the induction hypothesis and it follows that A is false in both I and $\Phi_P \uparrow (\beta \quad)$. $+1$ ■

6.5.4 Theorem Let P be Φ-accessible with unique supported model M. Then M is minimal as a two-valued model.

Proof: Let $\quad M$ be a model of P, and let l be the canonical level mapping of P wrt. Φ. Assume that there exists some $A \in M \setminus \quad$. Without loss of generality, we can assume that A is chosen such that $l(A)$ is minimal. By Definition 5.0.4 we obtain that there is a clause $A \leftarrow B_1, \ldots, B_k, \neg B_{k+1}, \ldots, \neg B_m$ in ground(P) with head A and $l(B_i) \quad (A)$ for all atoms B_i in the body. Since $B_{k+1}, \ldots, B_m \in M$, we $<l$
obtain $B_{k+1}, \ldots, B_m \in \quad$. By minimality of $l(A)$ we also obtain $B_1, \ldots, B_k \in \quad$. Now, since $\quad$ is a model of P, we must have $A \in \quad$ which is a contradiction to our assumption. ■

6.5.5 Program Theorem 6.5.4 cannot be generalized to all programs with unique supported models: the program

$$q \leftarrow p$$
$$p \leftarrow p, q$$
$$p \leftarrow \neg p, \neg q$$

has a unique supported model $\{p, q\}$, but $\{q\}$ is also a model (though not supported), and so $\{p, q\}$ is not minimal as a two-valued model.

Not also that for Φ^*-accessible programs the unique supported model is in general not least as a two-valued model as can be seen from the program consisting of the single clause $p \leftarrow q$.

6.5.6 Theorem The definite programs in $[\Phi]$ are exactly the definite programs with unique supported models.

Proof: This follows immediately from [Fit85, Proposition 7.3]: for a definite program P with least fixed point (I^+, I^-) of Φ_P, both I^+ and $B_P \setminus I^-$ are fixed points of the single-step operator T_P, and in fact I^+ is the least and $B_P \setminus I^-$ is the greatest supported model of P. Since P has only one supported model we obtain $I^+ = B_P \setminus I^-$ and therefore $P \in [\Phi]$. ■

6.6 Summary and Further Work

We have provided alternative characterizations of the classes of programs studied in Chapter 5, using operators on different three-valued logics. These logics turn out to be very closely related, and the novelty of this approach lies in the fact that the truth value *undefined* is employed in order to mirror aspects of the programs which are denotational, and not operational.

With this approach it was possible to characterize acceptable programs, i.e. programs which are terminating under SLDNF-resolution, and it is obvious to ask whether this approach can be carried over to termination analysis with respect to other resolution methods, or to other semantics which are based on many-valued logics, as referred to in the introduction of this chapter.

Chapter 7

Stable Model Semantics

The stable model semantics and the supported model semantics share the property that a program may have several meanings under these semantics, which is not the case under other semantics such as the well-founded [GRS91] or the weakly perfect model semantics [PP90]. The ambiguity of the stable model semantics, however, which at first sight seems to be an undesirable feature of it, has been put to use in a programming paradigm called *answer set programming*, which has currently been implemented in several forms, see [MT99] for an overview.

Stable models are always supported but not vice versa, so the stable model semantics can be viewed as a refinement of the supported model semantics. In this chapter, we will discuss some issues relating the two, and an application of the multivalued Kleene theorem 2.4.6.

In Section 7.1, we employ our results on Φ^*-accessible programs and a theorem due to [Fag91] in order to describe a class of programs for which their stable and their supported models coincide. Section 7.2 concerns the stable model semantics for disjunctive programs and how to relate it to the non-disjunctive case. Finally, in Section 7.3, we apply Theorem 2.4.6 in order to obtain stable models for a certain class of extended disjunctive programs, related to [KM98].

In this chapter, we will work over Herbrand interpretations only.

We will first give some preliminary definitions and results that will be needed in presenting our own results; they can all be found in [GL91, KM98], and in [GL88] for the non-disjunctive case. For most of this chapter, we will work with disjunctive programs, so we will shortly introduce them and their stable model semantics.

7.0.1 Definition Let $\mathtt{Lit}$ be the set of all ground literals in a first-order language $\mathcal{L}$. A *rule* r is an expression of the form

$$(L_1 \vee \qquad \vee L_n \leftarrow L_{n+1} \qquad\qquad L_m \quad \mathtt{not}\ L_{m+1} \qquad\qquad \mathtt{not}\ L_k)$$

where $L_i \in \mathtt{Lit}$ for each i. Rules are usually written as

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

Given such a rule r, we set $\mathtt{Head}(r) = \{L_1, \ldots, L_n\}$, $\mathtt{Pos}(r) = \{L_{n+1}, \ldots, L_m\}$ and $\mathtt{Neg}(r) = \{L_{m+1}, \ldots, L_k\}$. A rule r is said to be *disjunctive* if $n \geq 2$, and *non-disjunctive* otherwise. An *extended disjunctive program* is a countable set of disjunctive rules. If all the rules are non-disjunctive, the program is said to be *non-disjunctive*. The term *extended* refers to the use of two kinds of negation, one being classical negation, occurring in the literals of the clause, the other one being the negation `not`, which can be interpreted as negation as failure.

As an example of an extended disjunctive program we recall a version of the famous "Tweety" scenario.

7.0.2 Program

$$
\begin{aligned}
\mathrm{flies}(X) &\leftarrow \mathrm{bird}(X), \mathtt{not}\,\mathrm{penguin}(X)\\
\mathrm{abnormal}(X), \mathrm{flies}(X) &\leftarrow \mathrm{bird}(X)\\
\neg\,\mathrm{flies}(X) &\leftarrow \mathrm{penguin}(X)\\
\mathrm{bird}(X) &\leftarrow \mathrm{penguin}(X)\\
\mathrm{penguin}(\mathrm{tweety}) &\leftarrow\\
\mathrm{bird}(\mathrm{bob}) &\leftarrow
\end{aligned}
$$

The intended meaning of this program is that tweety is a penguin and a bird, does not fly, and is abnormal. But bob is a bird which does fly, since there is no evidence that bob is a penguin. Also, we have no evidence that bob is abnormal. This meaning is captured in the stable model semantics, introduced below.

Note that if P is a normal logic program, then ground(P) is an extended disjunctive logic program, which is in fact non-disjunctive and contains only one kind of negation. Since negation, $\neg$, in the case of normal logic programs can be understood from a procedural point of view as negation as failure, we interpret the occurrence of each negation $\neg$ in ground(P) as an instance of `not`. So ground(P), viewed as an extended disjunctive program, is non-disjunctive and contains only the negation `not`, so that all literals occurring in ground(P) are from this point of view in fact positive, i.e. atoms. As is customary in the literature, we will continue to use the symbol $\neg$ in this case to indicate negation as failure. Note that we assume that extended disjunctive programs are already given in ground form, while non-disjunctive programs may contain variable symbols. The identification of a program P with ground(P) does not pose any difficulties in the context of our discussion.

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

1. If X contains complementary literals, then $X = \texttt{Lit}$.
2. X is closed by rules in Π.

We denote the set of answer sets of Π by $\alpha(\Pi)$. If Π is non-disjunctive, then $\alpha(\Pi)$ is a singleton set, i.e. Π has only one answer set. However, if Π is disjunctive, then $\alpha(\Pi)$ may contain more than one element.

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

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

The program transformation $(\Pi, X) \to \Pi^X$ is called the *Gelfond-Lifschitz transformation* of Π with respect to X.

It is clear that the program Π^X does not contain `not` and therefore $\alpha\left(\Pi^X\right)$ is defined. We say that X is an *answer set* or *stable model* of Π if $X \in \alpha\left(\Pi^X\right)$. So, answer sets are fixed points of the operator GL introduced by Gelfond and Lifschitz in [GL91], where $\mathrm{GL}(X) = \alpha\left(\Pi^X\right)$. We note that the operator GL is in general not monotonic, and call it the *Gelfond-Lifschitz operator*.

In the current and the following chapter, we will also make slight use of the well-founded semantics, and we refer to [GRS91] for definitions and preliminary results.

7.1 Unique Supported and Stable Models

Since there exist many different semantics for logic programs, it is natural to ask when these semantics coincide. We will see in Theorem 8.2.3, that Φ-accessible programs are well-behaved from this point of view since all major semantics turn out to be the same for these programs. In this section, we will investigate a condition, in the non-disjunctive case, under which the stable models of a program are exactly the supported models of the program.

7.1.1 Proposition There is a program P which has a unique supported model but no stable model, and whose well-founded model is not total.

Proof: Consider the following program P:

$$\begin{aligned} p &\leftarrow p \\ p &\leftarrow \neg p \end{aligned}$$

We obtain $T_P(\{p\}) = \{p\}$ and $T_P(\emptyset) = \{p\}$, so $\{p\}$ is the unique supported model of P. However, the Gelfond-Lifschitz transformation using $\{p\}$ deletes the second clause and keeps the first. The resulting program has minimal model $\emptyset$, so $\{p\}$

is not a stable model. Since totality of the well-founded model implies that the well-founded model is stable [GRS91], we obtain that P does not have a total well-founded model. ■

We define well-supported Herbrand models following [Fag91, Fag94].

7.1.2 Definition An interpretation I of a program P is called *well-supported* if there exists a strict well-founded partial ordering $\prec$ on I such that for any atom $A \in I$ there exists a (ground) clause $A \leftarrow B_1, \dots, B_n, \neg C_1, \dots, \neg C_m$ such that $I \models B_1 \quad B_n \wedge \neg C_1 \quad \wedge \neg C_m$ and $B_i \prec A$ for each $i = 1, \dots, n$.

The following theorem was given in [Fag91, Theorem 2.1].

7.1.3 Theorem For a normal logic program P, the well-supported models of P are exactly the stable models of P.

Given a program P, we will denote by P' the program which is obtained from P as follows: P' is the set of all clauses $A \leftarrow A_1, \dots, A_n$ for which there is a clause $A \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m$ in P. Thus P' denotes the program which is obtained by omitting all negative literals in all the clauses in P, and we note that P' is definite.

We can now characterize a class of programs for which stable and supported models coincide. Recall that all stable models are supported.

7.1.4 Theorem Let P be a program such that P' is Φ^*-accessible. Then the supported models of P are exactly the stable models of P.

Proof: Let M be a supported model of P. We show that M is well-supported.

(1) M is a supported model of the Gelfond-Lifschitz transformation P^M of P with respect to M. In order to show this, let $A \leftarrow \texttt{body}$ be a clause in P^M, and assume that $\texttt{body}$ is true in M. Then the body of a corresponding clause in ground(P) is also true with respect to M by definition of P^M, and hence A is true with respect to M. So M is a model of P^M. To show supportedness, assume that $A \in M$. Then there is a clause $A \leftarrow \texttt{body}$ in P with $M \models \texttt{body}$. By definition of P^M we obtain that there is a corresponding clause in P^M whose body is true in M. So M is supported as a model of P^M.

(2) Since P' is Φ^*-accessible, it has a unique supported model . We show that M . Assume that this is not the case, i.e. that there is $A \in M \setminus$ with $l(A)$ minimal. Since M is a supported model of P^M, we know that there is a clause $A \leftarrow \texttt{body}$ in P^M with $M \models \texttt{body}$. But $\texttt{body}$ is also the body of a clause in P' with head A. So by Φ^*-accessibility of P', and since $A \in$ by assumption, there exists a literal B in $\texttt{body}$ with $l(B)$ $l(A)$ and $\models B$, and since P' is definite, we obtain $B \in M$ and $B \in$ which contradicts minimality of $l(A)$ in our choice of A. So M .

(3) We show now that M is well-supported as a model of P. Let $A \in M$. Since M is a supported model of P there exists a clause $A \leftarrow B_1, \dots, B_n \neg C_1, \dots, \neg C_m$ in ground(P) such that the body of this clause is true in M. From the inclusion

M it follows that $B_1, \ldots, B_n \in$. Now since P' is Φ^*-accessible we obtain $l(A) > l(B_i)$ for all $i = 1, \ldots, n$. Therefore, the strict ordering $\prec$ on M defined by $B \prec C$ if and only if $l(B)$ (C) establishes that the model M is well-supported. $<l$

■

The result in Theorem 7.1.4 cannot be generalized by replacing Φ^* with Φ: there exists a program P such that P' is Φ-accessible and such that P has a supported model which is not a stable model. In order to see this, let P be the program given in the proof of Proposition 7.1.1. Then P' has a unique supported model $M = \{p\}$ and is Φ-accessible. So M is indeed a supported model of P but not a stable model of P.

7.2 Stable Models and Supported Models in the Disjunctive Case

We study stable and supported models in the disjunctive case. In particular, we will provide a framework for casting disjunctive programs into non-disjunctive ones, and study relationships between the models before and after the transformation. We will work with disjunctive logic programs, i.e. with extended disjunctive programs where all literals occurring in the program are in fact positive, i.e. atoms. Moreover, `not` will be taken to mean classical negation, $\neg$. One immediate effect of this imposition that $\texttt{Head}(r)$ can only contain positive literals (whether or not the restriction on `not` is imposed) is to restrict the elements of an answer set to be positive literals also, as shown by the following lemma.

7.2.1 Lemma Suppose that the head of each clause in a disjunctive program Π contains only positive literals. Then any answer set for Π contains only positive literals.

Proof: Suppose that X is a set of literals which is closed by rules in Π^Z for some $\in 2^{\texttt{Lit}}$. Let Y denote the set which results by removing from X all the negative literals in X. Then Y is closed by rules in Π^Z. To see this, suppose that $r \in \Pi$ and that $\texttt{Pos}(r)$ Y is true. Then $\texttt{Pos}(r)$ X is also true, and so $\texttt{Head}(r) \cap Y = \texttt{Head}(r) \cap X = \emptyset$.

Therefore, by minimality, an answer set of Π can only contain positive literals.

■

Notice that this lemma makes redundant the condition 1. concerning complementary literals in the first part of the Definition 7.0.1 of an answer set.

Thus, for the rest of this section, the most general form of rule r that we shall consider in this section is the following

$$A_1, \ldots, A_n \leftarrow B_{n+1}, \ldots, B_m, \neg B_{m+1}, \ldots, \neg B_k,$$

where all A_i, B_j are atoms. Therefore, we have $\texttt{Head}(r) = \{A_1, \ldots, A_n\}$, $\texttt{Pos}(r) = \{B_{n+1}, \ldots, B_m\}$ and $\texttt{Neg}(r) = \{B_{m+1}, \ldots, B_k\}$.

In fact, the members of the class of disjunctive programs thus defined are precisely the disjunctive databases considered in [Prz88]. We will continue to use the notation Π for a typical disjunctive program even with this restriction in place. Hence, Π denotes a possibly infinite set of rules of the sort just described.

Normal Derivatives of Disjunctive Logic Programs

The Lemma 7.2.1 focuses attention on the sets of positive ground literals in the first order language $\mathcal{L}$ underlying Π i.e. on the power set I_Π of the Herbrand base B_Π of Π. We intend to relate answer sets to supported models of normal logic programs associated with Π, and Lemma 7.2.1 will assist us in doing this. Therefore, typical elements of I_Π will be denoted either by I or by X, depending on the context. The first step in the direction we want to go is provided by the following definition, and it will be convenient to write a typical rule r in Π in the form $H_r \leftarrow \mathtt{body}_r$.

7.2.2 Definition Suppose that Π is a disjunctive logic program. The *single-step operator T_Π associated with* Π is the multivalued mapping from I_Π to the power set 2^{I_Π} of I_Π defined by: $J \in T_\Pi(I)$ if and only if the following conditions are satisfied.

(i) For each rule $H_r \leftarrow \mathtt{body}_r$ in Π such that $I \models \mathtt{body}_r$, there exists an A in H_r such that $A \in J$.

(ii) For all $A \in J$, there exists a rule $H_r \leftarrow \mathtt{body}_r$ in Π such that $I \models \mathtt{body}_r$ and A belongs to H_r.

Notice that this definition reduces to the usual definition of the single-step operator T_P in case that Π is a normal logic program P.

7.2.3 Theorem Suppose that Π is a disjunctive logic program. Then we have $I \in T_\Pi(I)$, i.e. I is a fixed point of T_Π, if and only if the following conditions are satisfied.

(a) I is a model for Π, i.e. for every rule $H_r \leftarrow \mathtt{body}_r$ in Π such that $\mathtt{body}_r$ is true with respect to I, we have that H_r is also true with respect to I.

(b) For every $A \in I$, there is a rule $H_r \leftarrow \mathtt{body}_r$ in Π such that $\mathtt{body}_r$ is true with respect to I and $A \in H_r$.

By analogy with the non-disjunctive case, we call an interpretation I (i.e. an element of I_Π) which fulfills condition (b) above a *supported interpretation*. Thus, $I \in T_\Pi(I)$ if and only if I is a supported model for Π.

Proof: Suppose that $I \in T_\Pi(I)$ and let $H_r \leftarrow \mathtt{body}_r$ be a rule in Π such that $\mathtt{body}_r$ is true with respect to I. For (a), it remains to show that there is an atom A in H_r such that $A \in I$, which is the case by condition (i) of Definition 7.2.2. Condition (b) follows directly from (ii) of Definition 7.2.2.

Conversely, suppose that conditions (a) and (b) are satisfied by I. We have to show that $I \in T_\Pi(I)$, i.e. that conditions (i) and (ii) of Definition 7.2.2 are satisfied for $I = J$. Both however follow directly from conditions (a) and (b), respectively. ∎

We study next how to derive a normal program from a disjunctive one.

7.2.4 Definition Suppose that Π is a disjunctive logic program. A *normal derivative* P of Π is defined to be a (ground) normal logic program P consisting of possibly infinitely many clauses which satisfies the following conditions.

(a) For every rule $H_r \leftarrow \mathtt{body}_r$ in Π there exists a clause $A \leftarrow \mathtt{body}_r$ in P such that A belongs to H_r.

(b) For every clause $A \leftarrow \mathtt{body}_r$ in P there is a rule $H_r \leftarrow \mathtt{body}_r$ in Π such that A belongs H_r.

Note that condition (b) simply states that all clauses in P have to be derived from rules in Π by condition (a).

7.2.5 Theorem Let Π be a disjunctive logic program and let $I \in I_\Pi$. Then $J \in T_\Pi(I)$ if and only if $J = T_P(I)$ for some normal derivative P of Π.

Proof: Let P be a normal derivative of Π and suppose that $J = T_P(I)$. We have to show that $J \in T_\Pi(I)$ i.e. that J satisfies conditions (i) and (ii) of Definition 7.2.2.

For (i), let $H_r \leftarrow \mathtt{body}_r$ be a rule in Π such that $\mathtt{body}_r$ is true with respect to I. By condition (a) of the previous definition, there exists a clause $A \leftarrow \mathtt{body}_r$ in P such that A belongs to H_r. By definition of T_P, we have $A \in J$ as required.

For (ii), let A be in J. Then there exists a clause $A \leftarrow \mathtt{body}$ in P such that $\mathtt{body}$ is true with respect to I. By condition (b) of the previous definition, there exists a rule $H \leftarrow \mathtt{body}$ in Π such that A belongs to H as required.

Conversely, suppose that $J \in T_\Pi(I)$ i.e. that J satisfies conditions (i) and (ii) of Definition 7.2.2. We have to show that there exists a normal derivative P of Π such that $J = T_P(I)$. To do this, we define the ground normal program P as follows.

(1) Let $H_r \leftarrow \mathtt{body}_r$ be a rule in Π such that $\mathtt{body}_r$ is true with respect to I. Then by condition (i) there is an atom A in H_r such that $A \in J$. Let P contain all clauses $A \leftarrow \mathtt{body}_r$ for such A.

(2) For every rule $H_r \leftarrow \mathtt{body}_r$ in Π such that $\mathtt{body}_r$ is not true with respect to I, we choose an atom A in H_r arbitrarily. Let P contain all clauses $A \leftarrow \mathtt{body}_r$ thus defined.

(3) P contains only clauses defined by (1) and (2).

Obviously, P is a normal derivative of Π.

Now let $A \in J$. Then by (1) there exists a clause $A \leftarrow \mathtt{body}$ in P such that $\mathtt{body}$ is true with respect to I. Consequently, $A \in T_P(I)$. Conversely, let $A \in T_P(I)$. Then there is a clause $A \leftarrow \mathtt{body}$ in P such that $\mathtt{body}$ is true with respect to I. By (1) and (3) there exists a rule $H \leftarrow \texttt{body}$ in Π such that A belongs to H, and by (1) again, we obtain $A \in J$ as required. ■

The previous theorem allows us to conclude the existence of supported models for any given disjunctive program Π provided any normal derivative of Π has such a model. In particular, if any normal derivative of Π is acceptable, or locally hierarchical, or locally stratified[1], or definite, then Π has at least one supported model. Conversely, if a given disjunctive program Π has a supported model, there exists a normal derivative of Π which has a supported model. This fact is important from our point of view since we are focussing on normal derivatives of Π in the belief that they simplify the study of Π.

A *disjunctive database* Π is a finite disjunctive logic program consisting of $n_\Pi \in \mathbb{N}$ (ground) rules. We call n_Π the *order* of Π.

7.2.6 Proposition Let Π be a disjunctive database of order $n_\Pi = n \in \mathbb{N}$ consisting of the rules $r_1, r_2, \ldots, r_n$. For every $k \in \{1, \ldots, n\}$, let d_k denote the number of disjunctions occurring in the head of r_k. Then Π has at most $\prod_{k=1}^{n}\left(2^{d_k} - 1\right)$ normal derivatives. Therefore, for any $I \in I_\Pi$ we have $|T_\Pi(I)| \leq \prod_{k=1}^{n_\Pi}\left(2^{d_k} - 1\right)$.

Proof: Let r_k be a rule in Π. Every normal derivative P of Π contains at least one and at most d_k clauses generated by r_k. Consequently, there are $\;^{d_k}_{m=1}\binom{m}{d_k} = \left(\;^{d_k}_{m=0}\binom{m}{d_k}\right) - \binom{}{d_k} = 2^{d_k} - 1$ possibilities for clauses in P derived from r_k, and the first statement in the conclusion follows immediately from this. The second part of the conclusion now follows from Theorem 7.2.5. ■

For any disjunctive database which happens to be a normal logic program, the bound in the previous corollary turns out to be 1, so that this bound is sharp.

Normal Derivatives and the Answer Set Semantics

We now return to answer set semantics, and the final results of this section bring together the ideas developed thus far by relating answer sets of Π and supported models of normal derivatives of Π.

7.2.7 Theorem Suppose that Π is a disjunctive logic program in which $\texttt{Head}(r)$ contains only positive literals for each rule $r \in \Pi$, and in which $\texttt{not}$ denotes classical negation. Then given an answer set $X \in 2^{\texttt{Lit}}$ for Π, there is a normal derivative P of Π such that $T_P(X) = X$.

Proof: We have $X \in \alpha(\Pi^X)$. Consider Π^X and the following normal derivative P of Π which we construct by reference to the step by step construction of Π^X. Let r be a rule in Π and suppose for ease of notation that r takes the form $H_r \leftarrow \texttt{body}_r$.

First, suppose that $\texttt{Neg}(r) \cap X = \emptyset$, so that $r \in \Pi^X$. We choose an atom, A say, from the head H_r of r arbitrarily and include the clause $A \leftarrow \texttt{body}_r$ in P. Since

[1]Cf. Chapter 8.

$\mathtt{Neg}(r) \cap X = \emptyset$ we see that $X \models \mathtt{body}_r$, and therefore this clause contributes nothing to $T_P(X)$.

Now suppose that $\mathtt{Neg}(r) \cap X = \emptyset$. Then the rule r' belongs to Π^X, where r' is defined by $\mathtt{Head}(r') = \mathtt{Head}(r)$, $\mathtt{Pos}(r') = \mathtt{Pos}(r)$ and $\mathtt{Neg}(r') = \emptyset$. Since X is an answer set for Π^X, we have the statement $\mathtt{Pos}(r') \quad X \Rightarrow \mathtt{Head}(r') \cap X = \emptyset$ holding true. The first subcase of this case is when $\mathtt{Pos}(r') \quad X$. Again, we select an atom A in $\mathtt{Head}(r') = \mathtt{Head}(r)$ arbitrarily and include the clause $A \leftarrow \mathtt{body}_r$ in P. Since $\mathtt{Pos}(r) = \mathtt{Pos}(r') \quad X$, we have $X \models \mathtt{body}_r$ once more. Therefore, this clause also contributes nothing to $T_P(X)$.

Finally, consider the subcase of the previous case in which $\mathtt{Pos}(r') \quad X$, so that $\mathtt{Pos}(r) = \mathtt{Pos}(r') \quad X$. For each atom $A \in \mathtt{Head}(r') \cap X = \mathtt{Head}(r) \cap X$ include the clause $A \leftarrow \mathtt{body}_r$ in P, not including repetitions of this clause. Since $\mathtt{Pos}(r) \quad X$ and $\mathtt{Neg}(r) \cap X = \emptyset$, we have $X \models \mathtt{body}_r$. Thus, $T_P(X)$ includes all the $A \in \mathtt{Head}(r) \cap X$ for each rule r such that $\mathtt{Pos}(r) \quad X$. Therefore, we have $T_P(X) \quad X$, and P is a normal derivative of Π by construction. Thus, it remains to show that $T_P(X) = X$.

Suppose it is the case that $T_P(X) \subset X$ i.e. that there is an $x \in X$ such that for each rule r in Π^X with $\mathtt{Pos}(r) \quad X$ we have $x \in X \cap \mathtt{Head}(r)$. We show that this supposition leads to the contradiction that $Y = X \setminus \{x\} \subset X$ is an answer set for Π^X. Indeed, if r is a rule in Π^X such that $\mathtt{Pos}(r) \quad Y$, then $\mathtt{Pos}(r) \quad X$ and so $\mathtt{Head}(r) \cap Y = \mathtt{Head}(r) \cap X = \emptyset$. Thus, Y is closed by rules in Π^X. But this contradicts the minimality of X and concludes the proof. ■

As an immediate corollary of our results, we can recover the result of [GL91] that an answer set for Π is a model for Π (and hence the name answer set semantics or *stable model semantics*).

7.2.8 Corollary Suppose that Π is a disjunctive logic program. Then any answer set X for Π is a model for Π.

Proof: By Theorem 7.2.7, there is a normal derivative P of Π such that $T_P(X) = X$. Therefore, we have $X \in T_\Pi(X)$ by Theorem 7.2.5. It now follows that X is a supported model for Π by Theorem 7.2.3. ■

The following result is a first step towards a converse of Theorem 7.2.7.

7.2.9 Proposition Suppose that Π is a disjunctive logic program which satisfies the hypothesis of Theorem 7.2.7. Suppose also that $X \in 2^{\mathtt{Lit}}$ and that P is a normal derivative of Π such that $T_P(X) = X$. Then X is closed by rules in Π^X.

Proof: Let $r' \in \Pi^X$ be an arbitrary rule. Then there is a rule r in Π of the form $H_r \leftarrow \mathtt{body}_r$ such that $\mathtt{Neg}(r) \cap X = \emptyset$, $\mathtt{Head}(r') = \mathtt{Head}(r)$ and $\mathtt{Pos}(r') = \mathtt{Pos}(r)$. Suppose that $\mathtt{Pos}(r') \quad X$. Then $\mathtt{Pos}(r) \quad X$ and therefore $X \models \mathtt{body}_r$, since $\mathtt{Neg}(r) \cap X = \emptyset$. But P is a normal derivative of Π and therefore there must be a clause in P of the form $A \leftarrow \mathtt{body}_r$, where $A \in \mathtt{Head}(r)$. By definition of the single-step operator T_P, we have $A \in T_P(X)$ and hence we have $A \in X$ since

$T_P(X) = X$. Therefore, $\mathtt{Head}(r') \cap X = \mathtt{Head}(r) \cap X = \emptyset$. Thus, X is closed by rules in Π^X as stated. ■

Proposition 7.2.9 raises the problem of characterizing those normal derivatives whose fixed points are answer sets for Π. Indeed, the same problem can be put for all the other semantics which have been proposed for disjunctive programs and databases.

7.3 Signed Semi-disjunctive Programs

As already mentioned, the multivalued Knaster-Tarski theorem 2.1.4 was applied in [KM98] in order to find answer sets for a certain class of extended disjunctive programs, see Lemma 7.3.2 and Theorem 7.3.3 below. In this section, we will define a subclass of these programs to which the multivalued Kleene theorem 2.4.6 can be applied instead.

Recall, that the operator GL is in general not monotonic. However, for non-disjunctive programs it is *antimonotonic* in that we have $\mathrm{GL}(X) \supseteq \mathrm{GL}(Y)$ whenever X Y. This fact is used in order to obtain a monotonic operator by applying the operator GL twice. For this purpose, we partition a given program, if possible, into two suitable subprograms, following [KM98].

7.3.1 Definition An extended disjunctive logic program Π is said to be *signed* if there exists $S \in 2^{\mathtt{Lit}}$, called a *signing*, such that every rule $r \in \Pi$ satisfies one of the following conditions.

1. If $\mathtt{Neg}(r) \cap S$ is empty, then $\mathtt{Head}(r)$ S and $\mathtt{Pos}(r)$ S. Let Π_S be the subprogram of Π consisting of those rules which satisfy this condition.

2. If $\mathtt{Neg}(r) \cap S$ is not empty, then $\mathtt{Head}(r) \cap S = \mathtt{Pos}(r) \cap S = \emptyset$ and $\mathtt{Neg}(r)$ S. Let $\Pi_{\bar{S}}$ be the subprogram of Π consisting of those rules which satisfy this condition, where $\bar{S}$ denotes the set $\mathtt{Lit} \setminus S$.

Clearly, the programs Π_S and $\Pi_{\bar{S}}$ are disjoint and $\Pi = \Pi_S$ $\Pi_{\bar{S}}$. A signed program Π is said to be *semi-disjunctive* if there exists a signing S such that Π_S is non-disjunctive.

We borrow from [KM98] that, for signed semi-disjunctive programs, the operator $T : 2^{\bar{S}} \to 2^{2^{\bar{S}}}$ defined by

$$T(X) = \alpha\left(\Pi_{\bar{S}}^{\alpha\left(\Pi_S^X\right)}\right)$$

is monotonic with respect to the ordering $\supseteq$ which is the dual of the order of subset inclusion, . In fact, for the remainder of this section we will be concerned with decreasing orbits, and ω-continuity with respect to decreasing orbits etc. So, let us note that $2^{\mathtt{Lit}}$ is a complete lattice with respect to , and therefore the ordering $\supseteq$ on $2^{\texttt{Lit}}$ turns this set into an ω-cpo (with bottom element). Since it is natural to think of the ordering on $2^{\texttt{Lit}}$, rather than its dual, the notions and results of this section will be formulated with respect to . But, in fact, we will later on apply the dual version of the multivalued Kleene theorem 2.4.6, where the notions of monotonicity, ω-continuity and ω-cpo will be taken to mean the duals of the corresponding notions introduced in Section 2.4, see for example Lemma 7.3.2.

The following lemma, [KM98, Lemma 2], establishes the dual of the hypothesis (2.1) on T which was used in Theorem 2.1.4.

7.3.2 Lemma With the notation already established, let Π be a signed semi-disjunctive program, let (X_β) be a decreasing orbit of T in $2^{\bar{S}}$ and let X denote ${}_\beta X_\beta$. Then there exists $\bar{S}$ such that $\in T(X)$ and X.

From this lemma, it follows by the multivalued Knaster-Tarski theorem 2.1.4 that the operator T has a fixed point. The proof of the next theorem from [KM98] was based on this observation.

7.3.3 Theorem Let Π be a signed semi-disjunctive program which is *safe*[2] with respect to the partition $(\Pi_S, \Pi_{\bar{S}})$, where S is a signing for which Π_S is non-disjunctive. Then Π has a consistent answer set i.e. an answer set which does not contain any complementary literals.

The proof of this result utilizes only the single fact from Lemma 7.3.2 that a fixed point of T can be found (by applying Theorem 2.1.4). So, if a fixed point of T can be found by other means, the proof of Theorem 7.3.3, as given in [KM98], is still valid.

Now, if Π is a program as in Theorem 7.3.3 and, in addition to this, T is ω-continuous (using the notion dual to the one from Definition 2.4.5), then we obtain the fixed point of T from the proof of Theorem 2.4.6 using no more than ω iterations. We will see that a finiteness condition together with an acyclicity condition suffices to achieve this.

7.3.4 Definition A program Π is said to be of *finite type* if, for each $L \in \texttt{Lit}$, the set of rules in Π with L in their head is finite[3]. A program Π is called *acyclic* if there is a (level) mapping $l : \texttt{Lit} \to \mathbb{N}$, such that $l(L) = l(\neg L)$ for each literal L and, for every rule r in Π and for all L in $\texttt{Head}(r)$ and all L' in $\texttt{Pos}(r)$ $\texttt{Neg}(r)$, we have $l(L) > l(L')$.

The condition on a program that it is of finite type was used in [Sed95] in order to establish Theorem 4.2.6 concerning continuity, in the atomic topology, of the immediate consequence operator of a normal logic program i.e. of a non-disjunctive program. Later on it was shown in [Sed97] that continuity in the

[2] This concept is defined in [KM98], but it will not be needed here.

[3] When working with non-ground programs, a sufficient condition to obtain this for the ground instantiation of the program is the absence of local variables. See also Example 7.3.8.

atomic topology is closely related to continuity in quasimetric spaces. Thus, in the light of Section 2.4, it is not surprising that programs of finite type make an appearance again in our present setting. Cf. also Definition 4.3.11.

We now inductively define the following sets for a signed semi-disjunctive program with signing S.

$$
\begin{aligned}
X \;&= \texttt{Lit},\\
Y_i &= \alpha\left(\Pi_S^{X_i}\right),\\
X_{i+1} &\in \alpha\left(\Pi_{\bar{S}}^{Y_i}\right) \text{ with } X_{i+1} \quad X_i,\\
X &= \bigcap_{i\in\mathbb{N}} X_i,\\
Y &= \quad\; Y_i. \\
& \quad {}_{i\in\mathbb{N}}
\end{aligned}
$$

Indeed, these sets are well-defined since Π_S, and therefore $\Pi_S^{X_i}$, is non-disjunctive for each i, and since the operator T, where $T(X_i) = \alpha \quad \Pi_{\bar{S}}^{\alpha\left(\Pi^{X_i}\right)}$ as above, is monotonic. With this notation, we have the following lemma.

7.3.5 Lemma Let Π be a signed semi-disjunctive program with signing S such that $\Pi_{\bar{S}}$ is of finite type. Then the following hold with respect to the ordering on $2^{\texttt{Lit}}$.

(i) The sequence X_i is decreasing. We set $X = \quad X_i$.

(ii) The sequence $\Pi_S^{X_i}$ of programs is increasing with respect to set-inclusion, and $\quad \Pi_S^{X_i} = \Pi_S^{X}$.

(iii) The sequence Y_i is increasing. We set $Y = \quad Y_i$.

(iv) The sequence $\Pi_{\bar{S}}^{Y_i}$ of programs is decreasing with respect to set-inclusion, and $\quad \Pi_{\bar{S}}^{Y_i} = \Pi_{\bar{S}}^{Y}$.

(v) $Y = \alpha\left(\Pi_S^X\right)$.

(vi) X is closed by rules in $\Pi_{\bar{S}}^{Y}$.

(vii) For each L in X, there is a rule r in $\Pi_{\bar{S}}^{Y}$ with $L \in \texttt{Head}(r)$ such that the following two conditions are satisfied.

 (vii.1) $\texttt{Pos}(r) \quad X$.

 (vii.2) For any literal $L' \in \texttt{Head}(r)$ with $L' = L$, we have $L' \in X$.

Proof: (i) This follows immediately from the definition of the X_i.
(ii) This follows from (i), (iii) follows from (ii), and (iv) follows from (iii).
(v) If $L \in Y$, then there is $i \quad \in \mathbb{N}$ such that $L \in Y_i = \alpha\left(\Pi_S^{X_i}\right)$ for all $i \geq i$.

Since the sequence $\Pi_S^{X_i}$ of programs is increasing with respect to set-inclusion and $\Pi_S^{X_i} \quad \Pi_S^X$ for each i, we obtain $L \in \alpha\left(\Pi_S^X\right)$ and therefore $Y \quad \alpha\left(\Pi_S^X\right)$. Now let r be a clause in Π_S^X. If $\texttt{Pos}(r) \quad Y$, then there is $i \in \mathbb{N}$ such that $\texttt{Pos}(r) \quad Y_i$. But each Y_i is closed by rules in $\Pi_S^{X_i}$ and $\Pi_S^{X_i}$ is non-disjunctive for each i, hence we obtain that $\texttt{Head}(r) \in Y_i$. So $\texttt{Head}(r) \in Y$ and it follows that Y is closed by rules in Π_S^X. Since answer sets of Π_S^X are sets which are minimally closed by rules in Π_S^X and since $Y \quad \alpha\left(\Pi_S^X\right)$, we obtain that $Y = \alpha\left(\Pi_S^X\right)$.
(vi) This was shown in [KM98].
(vii.1) Let $L \in X$ be a literal. We know that $L \in X_n$ for all n. But X_n is minimally closed by rules in $\Pi_{\bar{S}}^{Y_n}$, therefore we also know that, for each n, there must be a rule r in $\Pi_{\bar{S}}^{Y_n}$ with $L \in \texttt{Head}(r)$ and $\texttt{Pos}(r) \quad X_n$. Since $\Pi_{\bar{S}}$ is of finite type, we also know that there are only finitely many rules r in $\Pi_{\bar{S}}^{Y_n}$ with $L \in \texttt{Head}(r)$. But $\Pi_{\bar{S}}^{Y_i+1} \quad \Pi_{\bar{S}}^{Y_i}$ for all i, so it follows that there must be a rule r in $\Pi_{\bar{S}}^{Y}$ with $L \in \texttt{Head}(r)$ such that $\texttt{Pos}(r) \quad X_i$ for all i. Hence $\texttt{Pos}(r) \quad X$.
(vii.2) Let $r_1, \ldots, r_n$ be all the rules in $\Pi_{\bar{S}}^{Y}$ with $L \in \texttt{Head}(r_i)$ and $\texttt{Pos}(r_i) \quad X$, noting that $\Pi_{\bar{S}}^{Y}$ is of finite type so that there exist only finitely many such rules. There must now be a $j \ \in \mathbb{N}$ such that, for all $j \geq j$, we have that each r_i is a rule in $\Pi_{\bar{S}}^{Y_j}$ with $\texttt{Pos}(r_i) \quad X_j$ by (vii.1). Now, for each $i = 1, \ldots, n$, suppose that there is a literal $L_i = L$ in $\texttt{Head}(r_i)$ with $L_i \in X$. Then we have $L_i \in X_j$ for all $j \geq j$. It is now easy to see that $X_j \setminus \{L\}$ is closed by rules in $\Pi_{\bar{S}}^{Y_j}$, which contradicts the fact that X_j is minimally closed by rules in $\Pi_{\bar{S}}^{Y_j}$. ■

If the program $\Pi_{\bar{S}}$ additionally satisfies the acyclicity condition, then X is already a fixed point of T, as we show next.

7.3.6 Theorem Let Π be a signed semi-disjunctive program with signing S such that $\Pi_{\bar{S}}$ is of finite type and is acyclic. Let (X_n) be a decreasing ω-orbit of T in $2^{\bar{S}}$ and let $X = \quad_n X_n$. Then $X \in T(X)$.

Proof: We know from Lemma 7.3.2 that there is $\quad X$ with $\quad \in T(X)$. Assume $' = X \setminus \quad = \emptyset$. Since $\Pi_{\bar{S}}^{Y}$ is acyclic, there must be an $L \in \quad '$ of minimal level. But $L \in X$ so, by Lemma 7.3.5 (vii), there must be a rule r which satisfies conditions (vii.1) and (vii.2). By (vii.1) and minimality of the level of L, we obtain $\texttt{Pos}(r)$ and since is closed by rules in $\Pi_{\bar{S}}^{Y}$, there must be a literal $L' \in \texttt{Head}(r)$ with $L' \in$. But $\quad X$, so we obtain $L' \in X$ and $L' = L$ by (vii.2), and therefore $L \in$ contradicts $L \in$. ■

As already mentioned above, the proof of Theorem 7.3.3 now carries over directly from [KM98], so that each signed semi-disjunctive program which is safe with respect to the partition $(\Pi_S, \Pi_{\bar{S}})$, where S is a signing for which Π_S is non-disjunctive and $\Pi_{\bar{S}}$ is of finite type and acyclic, has a consistent answer set. From the proof of Theorem 7.3.3 together with Theorem 7.3.6, this answer set turns out to be $Y \quad X$, with notation as defined in the paragraph preceding Lemma 7.3.5. The novelty of this theorem lies in the fact that the answer set can be found by applying the operator T no more than ω times.

We conclude with two examples which show that the conditions of being acyclic and of finite type are indeed necessary. We will use the notation from Lemma 7.3.5.

7.3.7 Program Let Π be the ground instantiation of the following program, where x denotes a variable and 0 a constant.

$$\begin{aligned} p(x) &\leftarrow \texttt{not}\ q(x) \\ q(s(x)) &\leftarrow \texttt{not}\ p(x) \\ r(0) &\leftarrow q(x), \texttt{not}\ p(x) \end{aligned}$$

The program Π is signed with signing $S = \{p(s^n(0)) \mid n \in \mathbb{N}\}$ and is trivially semi-disjunctive. Note, however, that $\Pi_{\bar{S}}$ is not of finite type but is acyclic. We now make the following calculations:

$$\begin{aligned} X\ \ &= \texttt{Lit}, \\ Y\ \ &= \emptyset, \\ X_i &= \{r(0)\} \quad \{q(s^n(0)) \mid n \geq i\} \text{ for } i \geq 1, \\ Y_i &= \{p(s^n(0)) \mid n = 1, \dots, i\} \text{ for } i \geq 1. \end{aligned}$$

As expected, the set $X \ = \ {}_i X_i = \{r(0)\}$ is not a fixed point of T nor is $X \quad {}_i Y_i = \{r(0)\} \cup \{p(s^n(0)) \mid n \in \mathbb{N}\}$ an answer set of Π. However, taking $X_{\ +1} = T(X\ ) = \emptyset$, which is a fixed point of T, we obtain $\{p(s^n(0)) \mid n \in \mathbb{N}\}$ as answer set of Π.

The following example shows that the acyclicity condition on $\Pi_{\bar{S}}$ cannot be dropped.

7.3.8 Program Let Π be the ground instantiation of the following program, where x is a variable and a constant symbol 0 is added to the language underlying Π.

$$\begin{aligned} t(x) &\leftarrow t(x) \\ p(x) &\leftarrow \texttt{not}\ q(x) \\ q(s(x)) &\leftarrow \texttt{not}\ p(x) \\ r(x) &\leftarrow q(x), \texttt{not}\ p(x) \\ r(x) &\leftarrow r(s(x)), \texttt{not}\ t(x) \end{aligned}$$

The program Π is signed with respect to the signing $S = \{p(s^n(0)), t(s^n(0)) \mid n \in \mathbb{N}\}$ and is trivially semi-disjunctive. Note, however, that due to the last clause in the above program, $\Pi_{\bar{S}}$ is not acyclic but is of finite type. We now make the following calculations:

$$\begin{aligned} X\ \ &= \texttt{Lit}, \\ Y\ \ &= \emptyset, \\ X_i &= \{q(s^n(0)) \mid n \geq i\} \cup \{r(s^n(0)) \mid n \in \mathbb{N}\} \text{ for } i \geq 1, \\ Y_i &= \{p(s^n(0)) \mid n = 0, \dots, i-1\} \text{ for } i \geq 1. \end{aligned}$$

As expected, the set $X \quad = \quad {}_i X_i = \{r(s^n(0)) \mid n \in \mathbb{N}\}$ is not an answer set of $\Pi_{\bar{S}}$ nor is $X \qquad {}_i Y_i = \{r(s^n(0)) \mid n \in \mathbb{N}\} \cup \; \{p(s^n(0)) \mid n \in \mathbb{N}\}$ an answer set of Π. However, if we keep on iterating and calculate

$$
\begin{aligned}
Y_{\;+1} &= \alpha\left(\Pi_S^{X_\omega}\right) = \{p(s^n(0)) \mid n \in \mathbb{N}\}, \text{ and} \\
X_{\;+1} &= T(X \;\,) = \emptyset
\end{aligned}
$$

we obtain $X_{\;+1}$ as fixed point of T and $\{p(s^n(0))\}$ as answer set of Π.

7.4 Summary and Further Work

We have discussed relationships between the stable model semantics and the supported model semantics, and applied a fixed-point theorem from Chapter 1 to the Gelfond-Lifschitz operator for extended disjunctive programs. We note that our methods of obtaining normal programs from disjunctive ones as in Section 7.2 correspond to relationships between the multivalued fixed-point theorems from Chapter 2 and the respective single-valued theorems from Chapter 1.

Stable models can be understood in the framework of *default theories* due to R. Reiter, and are important for the currently emerging programming paradigm called *answer set programming*. Domain-theoretic investigations of the stable model semantics have been undertaken in [ZR97a, ZR97b, ZR98, RZ98], where disjunctive programs were treated using Smyth powerdomains instead of multivalued mappings. Relationships to the work presented in this chapter suggest themselves but remain to be worked out.

We finally note that there is a subtle difference between programs P and their ground instantiations ground(P). Every program P can be cast into a possibly infinite ground program by associating it with ground(P). However, a countably infinite ground program cannot in general be converted into a finite program containing variables. While this does not cause any restrictions concerning the denotational analysis of these programs, there is certainly a difference when talking about operational aspects, e.g due to the presence of floundering under SLDNF-resolution. We would also like to mention [Fer94], where classes of models are characterized in topological terms. This work is based on (possibly infinite) ground programs, and, due to our observations above, can not be carried over without modifications to the case of finite programs with variables.

Chapter 8

Perfect and Weakly Perfect Model Semantics

The perfect model semantics was proposed in [Prz88] as a suitable semantics for locally stratified programs, introduced below, which are a common generalization of both locally hierarchical and stratified programs [ABW88]. It turned out to be too restrictive, however, and the approach was generalized in [PP90] to the so-called weakly stratified programs, resulting in the weakly perfect model semantics.

In Section 8.1, we will study the perfect model semantics for locally stratified programs from an iterative point of view, inspired by an approach followed in [ABW88] for stratified programs. In Section 8.2, we investigate Φ-accessible programs from the point of view of the weakly perfect model semantics and show, that all major semantic approaches coincide for these programs.

We will work over Herbrand interpretations only.

8.1 Locally Stratified Programs

We first define stratified programs due to [ABW88].

8.1.1 Definition Let P denote a normal logic program. Then P is said to be *stratified* if there is a partition $P = P_1 \qquad P_m$ of P such that the following two conditions hold for $i = 1, \dots, m$:

(1) If a predicate symbol occurs positively in a clause in P_i, then its definition is contained within $\quad_{j \le i} P_j$.

(2) If a predicate symbol occurs negatively in a clause in P_i, then its definition is contained within $\quad_{j < i} P_j$.

We adopt the convention that the definition of a predicate symbol p occurring in P is contained in P_1 whenever its definition is empty. Thus, each predicate symbol occurring in P is defined but it may have empty definition; in particular, P_1 itself may be empty.

In order to treat non-monotonic operators, the *powers* of an operator T mapping a complete lattice into itself were defined in [ABW88] as follows:

$$\begin{aligned} T\!\uparrow\!0(I) &= I \\ T\!\uparrow\!(n \quad 1)(I) &= T(T\!\uparrow\!n(I)) \quad T\!\uparrow\!n(I) \\ T\!\uparrow\!\omega(I) &= \quad \overset{\infty}{\underset{n=0}{}} T\!\uparrow\!n(I). \end{aligned}$$

Of course, $T\!\uparrow\!n(I)$ is *not* equal to $T^n(I)$ unless T is monotonic and $I \quad T(I)$. Indeed, the sequence $(T\uparrow n(I))_n$ is always monotonic increasing. However, this concept can be used to construct a minimal supported model M_P for any stratified program P as follows: put $M \;\; = \emptyset$, $M_1 = T_{P_1}\!\uparrow\!\omega(M \;\;), \ldots, M_m = T_{P_m}\!\uparrow\!\omega(M_{m-1})$. Finally, let $M_P = M_m$. This construction is due to [ABW88].

We next define locally stratified programs due to [Prz88] which generalize both stratified and locally hierarchical programs.

8.1.2 Definition A normal logic program P is called *locally stratified* if there exists a level mapping $l : B_P \to \gamma$ for P such that for every clause $A \leftarrow A_1, \ldots, A_m, \neg B_1, \ldots, \neg B_n$ in ground(P) we have $l(A) \geq l(A_i)$ and $l(A) > l(B_j)$ for all i and j.

While the defining conditions for locally hierarchical programs prevent the occurrence of recursion, the conditions for locally stratified programs prevent only recursion through negation, hence allow to control the negation which occurs in the program, as we will see below, without restricting the use of recursion otherwise. In particular, each definite program is locally stratified.

We will now carry over the above mentioned treatment of stratified programs to the case of locally stratified programs.

8.1.3 Definition Let P denote a normal logic program and let $l : B_P \to \gamma$ denote a level mapping, where $\gamma > 1$. For each n satisfying $0 \quad n \leq \gamma$, let $P_{[n]}$ denote the set of all clauses in ground(P) in which only atoms A with $l(A)$ occur, and recall the notation $\mathcal{L}_n$ for the set of all atoms A of level $l(A)$ less than n. We define $T_{[n]} : \quad (\mathcal{L}_n) \to \quad (\mathcal{L}_n)$ by $T_{[n]}(I) = T_{P_{[n]}}(I)$. The mapping $T_{[n]}$ is called the *immediate consequence operator restricted at level* n.

Thus, the idea formalized by this definition is to "cut-off" at level n.

8.1.4 Construction Let P be a locally stratified program and let $l : B_P \to \gamma$ denote a level mapping, where $\gamma > 1$. We construct the transfinite sequence $(I_n)_{n\in\gamma}$ inductively as follows. For each $m \in \mathbb{N}$, we put $I_{[1,m]} = T_{[1]}^m(\emptyset)$ and set $I_1 = \quad {}^{\infty}_{m=0} I_{[1,m]}$. If $n \in \gamma$, where $n > 1$, is a successor ordinal, then for each $m \in \mathbb{N}$ we put $I_{[n,m]} = T_{[n]}^m(I_{n-1})$ and set $I_n = \quad {}^{\infty}_{m=0} I_{[n,m]}$. If $n \in \gamma$ is a limit ordinal, we put $I_n = \quad {}_{m<n} I_m$. Finally, we put $I_{[P]} = \quad {}_{n<\gamma} I_n$.

The main technical lemma we need is as follows. For its proof, which is by transfinite induction, it will be convenient to put $I_{[n,m]} = I_n$ for all $m \in \mathbb{N}$

whenever n is a limit ordinal; thus statement (b) in the lemma makes sense for all ordinals n.

8.1.5 Lemma Let P be a normal logic program which is locally stratified with respect to the level mapping $l : B_P \to \gamma$, where $\gamma > 1$. Then the following statements hold.

(a) The sequence $(I_n)_{n\in\gamma}$ is monotonic increasing in n.

(b) For every $n \in \gamma$, where $n \geq 1$, the sequence $(I_{[n,m]})$ is monotonic increasing in m.

(c) For every $n \in \gamma$, where $n \geq 1$, I_n is a fixed point of $T_{[n]}$.

(d) If $l(B)$ and $B \in I_n$, where $B \in B_P$, then for every $m \in \gamma$ with $n < m$ we have $B \in I_m$ and hence $B \in I_{[P]}$. In particular, if $l(B)$ and $B \in I_{[n+1,m]}$ for some $m \in \mathbb{N}$, then $B \in I_n$ and hence $B \in I_{[P]}$.

Proof: It is immediate from the construction that the sequence $(I_n)_{n\in\gamma}$ is monotonic increasing in n, and this establishes (a).

The main work is in establishing (b) and (c), which we treat simultaneously. To do this, we need to note the technical fact that, for each $n \in \gamma$, we can partition $P_{[n+1]}$ as $P_{[n]}$ $P(n)$, where $P(n)$ denotes the subset of ground(P) consisting of those clauses whose head has level n. Thus, $T_{[n+1]}(I) = T_{[n]}(I)$ $T_{P(n)}(I)$ for any $I \in I_P$; note that if $A \in T_{P(n)}(I)$, then $l(A) = n$.

Let (n) be the proposition, depending on the ordinal n, that $(I_{[n,m]})$ is monotonic increasing in m and that I_n is a fixed point of $T_{[n]}$. Suppose that (n) holds for all n α, where $\alpha \leq \gamma$ is some ordinal. We must show that (α) holds. Indeed, (1) holds since $P_{[1]}$ is a definite program and the construction of I_1 is simply the classical construction of the least fixed point of $T_{[1]}$, and therefore we may assume that $\alpha > 2$. It will be convenient to break up the details of the case when α is a successor ordinal into a sequence of steps.

Case 1. $\alpha = k$ 1 is a successor ordinal. Thus, (k) holds.

Step 1. We establish the recursion equations:

$$I_{[k+1,0]} = I_k$$
$$I_{[k+1,m+1]} = I_k \quad T_{P(k)}(I_{[k+1,m]})$$

and the first is immediate. Putting $m = 0$, we have $I_{[k+1,1]} = T_{[k+1]}(I_k) = T_{[k]}(I_k)$ $T_{P(k)}(I_k) = I_k$ $T_{P(k)}(I_k) = I_k$ $T_{P(k)}(I_{[k+1,0]})$, 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 $I_{[k+1,(m+1)+1]} = T_{[k+1]}(I_{[k+1,m+1]}) = T_{[k]}(I_{[k+1,m+1]})$ $T_{P(k)}(I_{[k+1,m+1]}) = T_{[k]}(I_k$ $T_{P(k)}(I_{[k+1,m]}))$ $T_{P(k)}(I_{[k+1,m+1]})$, and it suffices to show that $T_{[k]}(I_k$ $T_{P(k)}(I_{[k+1,m]})) = I_k$. So suppose that $A \in T_{[k]}(I_k$ $T_{P(k)}(I_{[k+1,m]}))$. Thus, there is a clause in $P_{[k]}$ of the form $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ where $A_1, \ldots, A_{k_1} \in I_k$ $T_{P(k)}(I_{[k+1,m]})$ and $B_1, \ldots, B_{l_1} \in I_k$ $T_{P(k)}(I_{[k+1,m]})$. But then level considerations and the hypothesis concerning P imply that $A_1, \ldots, A_{k_1} \in I_k$

and $B_1, \ldots, B_{l_1} \in I_k$. Therefore, $A \in T_{[k]}(I_k) = I_k$ and we have the inclusion $T_{[k]}(I_k \quad T_{P(k)}(I_{[k+1,m]})) \quad I_k$. The reverse inclusion is demonstrated in like fashion, showing that the second of the recursion equations holds with m replaced by $m \quad 1$ and hence, by induction on m, that it holds for all m.

Step 2. We have the inclusions $T_{P(k)}(I_k) \quad T_{P(k)}(I_k \quad T_{P(k)}(I_k)) \quad T_{P(k)}(I_k \quad T_{P(k)}(I_k \quad T_{P(k)}(I_k))) \ldots$

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 1 that $(I_{[k+1,m]})$, or $(I_{[\alpha,m]})$, is monotonic increasing in m. Since monotonic increasing sequences converge to their union in Q, and $I_{[k+1,m]}$ is an iterate of I_k, it now follows by Theorem 4.2.5 that I_{k+1} is a model for $P_{[k+1]}$.

Step 3. If $B \in B_P$ and $l(B) \quad$, then $B \in I_{k+1}$ if and only if $B \in I_k$.

Indeed, if $B \in I_k$, then it is clear from the recursion equations of Step 1 that $B \in I_{k+1}$. On the other hand, if $B \in I_k$, then it is equally clear from the recursion equations and level considerations that, for every $m \in \mathbb{N}$, $B \in I_{[k+1,m]}$ and hence that $B \in I_{k+1}$, as required.

Step 4. I_{k+1} is a supported model for $P_{[k+1]}$.

To see this, suppose that $A \in I_{k+1} = \ {}_{m=0}^{\infty} I_{[k+1,m]}$. Then there is $m \ \in \mathbb{N}$ such that $A \in I_{[k+1,m+1]} = T_{[k+1]}^{m+1}(I_k)$ for all $m \geq m$. 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 $l(B_j) \quad k$ for each j since P is locally stratified. Since $B_j \in I_{[k+1,m_0]}$, we now see from the recursion equations that $B_j \in I_k$. From the result in Step 3 we now deduce that, for each j, $B_j \in I_{k+1}$. 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, $I_{k+1} \quad T_{[k+1]}(I_{k+1})$ and therefore I_{k+1} is a supported model for $P_{[k+1]}$, or a fixed point of $T_{[k+1]}$, as required.

Thus, $\ (\alpha)$ holds when α is a successor ordinal.

Case 2. α is a limit ordinal.

In this case, it is trivial that $(I_{[\alpha,m]})$ is monotonic increasing in m. Thus, we have only to show that I_α is a fixed point of $T_{[\alpha]}$ i.e. a supported model for $P_{[\alpha]}$, and we show first that I_α is a model for $P_{[\alpha]}$. Let $A \in T_{[\alpha]}(I_\alpha)$. Then there is a clause $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ in $P_{[\alpha]}$ such that $A_1, \ldots, A_{k_1} \in I_\alpha$ and $B_1, \ldots, B_{l_1} \in I_\alpha$. Indeed, by the definition of $P_{[\alpha]}$ and the hypothesis concerning P, there is $n \ \alpha$ such that the clause $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ belongs to $P_{[n_0]}$. Since the sequence $(I_n)_{n \in \gamma}$ is monotone increasing and $I_\alpha = \ {}_{n<\alpha} I_n$, there is $n_1 \quad$ such that $A_1, \ldots, A_{k_1} \in I_{n_1}$ and $B_1, \ldots, B_{l_1} \in I_{n_1}$. Choosing $n_2 = \max\{n \ , n_1\}$, we have $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1} \in P_{[n_2]}$ and also $A_1, \ldots, A_{k_1} \in I_{n_2}$ and $B_1, \ldots, B_{l_1} \in I_{n_2}$. Therefore, on using the induction hypothesis we have $A \in T_{[n_2]}(I_{n_2}) = I_{n_2} \quad I_\alpha$. Hence, $T_{[\alpha]}(I_\alpha) \quad I_\alpha$, as required.

To see that I_α is supported, let $A \in I_\alpha$. By monotonicity of $(I_n)_{n \in \gamma}$ again and the identity $I_\alpha = \ {}_{n<\alpha} I_n$, there is a successor ordinal $n \ \geq 1$ such that $A \in I_n$ for all n such that $n \ \leq n \quad \alpha$. In particular, we have $A \in I_{n_0} = \ {}_{m=0}^{\infty} I_{[n_0,m]}$. Therefore, there is $m_1 \in \mathbb{N}$ such that $A \in I_{[n_0,m_1+1]} = T_{[n_0]}(T_{[n_0]}^{m_1}(I_{n_0-1}))$. Consequently, there is a clause $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ in $P_{[n_0]}$ such that $A_1, \ldots, A_{k_1} \in T_{[n_0]}^{m_1}(I_{n_0-1}) = I_{[n_0,m_1]} \quad I_{n_0} \quad I_\alpha$ and $B_1, \ldots, B_{k_1} \in I_{[n_0,m_1]}$. But $l(B_j) \quad\quad -1$ for each j and so no B_j belongs to I_{n_0-1} by Step 3 of the previous case. Therefore, by this step, no B_j belongs to I_{n_0} and by iterating this we see that, for every $m \in \mathbb{N}$, no B_j belongs to I_{n_0+m}. Therefore, no B_j belongs to I_α. Hence, we have $A \in T_{[n_0]}(I_\alpha) \quad T_{[\alpha]}(I_\alpha)$ or in other words that $I_\alpha \quad T_{[\alpha]}(I_\alpha)$, as required.

It now follows that $\quad (n)$ holds for all ordinals n, and this completes the proof of (b) and (c). In particular, we see that the recursion equations obtained in Step 1 hold for all ordinals k, and we record this fact in the corollary below. Indeed, all that is needed to establish these equations is the fact that each I_k is a fixed point of $T_{[k]}$, and to note that the proof just given shows also that $I_{[P]}$ is a fixed point of T_P. In turn, (d) of the lemma now follows from this observation by iterating Step 3.

The proof of the lemma is therefore complete. ■

It can be seen here, and it will be seen again later, that the importance of (d) is the control it gives over negation in the manner illustrated in the proof just given that I_{k+1} is a supported model for $P_{[k+1]}$. It is also worth noting that the construction produces a monotonic increasing sequence by means of a non-monotonic operator, and that Lemma 8.1.5 plays a role here similar to that played by [ABW88, Lemma 10].

8.1.6 Corollary Suppose the hypotheses of Lemma 8.1.5 all hold. Then:

(1) For all ordinals n and all $m \in \mathbb{N}$ we have the recursion equations

$$
\begin{aligned}
I_{[n+1,0]} &= I_n \\
I_{[n+1,m+1]} &= I_n \quad T_{P(n)}(I_{[n+1,m]}).
\end{aligned}
$$

(2) If P is in fact locally hierarchical, then for every ordinal $n \geq 1$ we have $I_{[n+1,m]} = I_n \quad T_{P(n)}(I_n)$ for all $m \in \mathbb{N}$, where $P(n)$ is defined as in the proof of Lemma 8.1.5, and therefore the iterates stabilize after one step.

Proof: That (1) holds has already been noted in the proof of Lemma 8.1.5. For (2), it suffices to prove that $T_{P(n)}(I_n) = T_{P(n)}(I_n \quad T_{P(n)}(I_n))$. So suppose therefore that $A \in T_{P(n)}(I_n \quad T_{P(n)}(I_n))$. Then there is a clause $A \leftarrow A_1, \ldots, A_{k_1}, \neg B_1, \ldots, \neg B_{l_1}$ in $P(n)$ such that $A_1, \ldots, A_{k_1} \in I_n \quad T_{P(n)}(I_n)$ and $B_1, \ldots, B_{k_1} \in I_n \quad T_{P(n)}(I_n)$. From these statements and by level considerations, we have $A_1, \ldots, A_{k_1} \in I_n$ and $B_1, \ldots, B_{k_1} \in I_n$. Therefore, $A \in T_{P(n)}(I_n)$ so that $T_{P(n)}(I_n \quad T_{P(n)}(I_n)) \quad T_{P(n)}(I_n)$. The reverse inclusion is established similarly to complete the proof. ■

Statement (2) of this corollary makes the calculation of iterates very easy to perform in the case of locally hierarchical programs.

8.1.7 Theorem Suppose that P is a normal logic program which is locally stratified with respect to the level mapping $l : B_P \to \gamma$. Then $I_{[P]}$ is a minimal supported model for P.

Proof: That $I_{[P]}$ is a supported model for P follows from the proof of Lemma 8.1.5, and so it remains to show that $I_{[P]}$ is minimal. To do this, we establish by transfinite induction the following proposition: "if $J \quad I_{[P]}$ and $T_P(J) \quad J$, then $I_n \quad J$ for all $n \in \gamma$, where $n \geq 1$", and this clearly suffices. Indeed, $T_{[1]}(J) \quad T_P(J) \quad J$ and therefore J is a model for $P_{[1]}$. But, as already noted in proving Lemma 8.1.5, I_1 is the least model for $P_{[1]}$ by construction, since $P_{[1]}$ is definite. Therefore, $I_1 \quad J$ and the proposition holds with $n = 1$.

Now assume that the proposition holds for all ordinals $n \quad \alpha$ for some ordinal $\alpha \in \gamma$, where $\alpha > 1$; we show that it holds with $n = \alpha$.

Case 1. $\alpha = k \quad 1$ is a successor ordinal, where $k > 0$.

We have $I_k \quad J$. We show by induction on m that $I_{[k+1,m]} \quad J$ for all m. Indeed, with $m = 0$ we have $I_{[k+1,0]} = I_k \quad J$. Suppose, therefore, that $I_{[k+1,m_0]} \quad J$ for some $m \;\; > 0$. Let $A \in I_{[k+1,m_0+1]} = T_{[k+1]}(T^{m_0}_{[k+1]}(I_k))$. Then 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 $A_1, \ldots, A_{k_1} \in T^{m_0}_{[k+1]}(I_k) = I_{[k+1,m_0]}$ and $B_1, \ldots, B_{l_1} \in I_{[k+1,m_0]}$. But $l(B_j) \qquad$ for each j. Applying Lemma 8.1.5 (d) we see that no B_j belongs to $I_{[P]}$ and consequently no B_j belongs to J because $J \quad I_{[P]}$. Since $I_{[k+1,m_0]} \quad J$ by assumption, we have $A_1, \ldots, A_{k_1} \in J$. Therefore, $A \in T_{[k+1]}(J) \quad T_P(J) \quad J$, and from this we obtain that $I_{[k+1,m_0+1]} \quad J$ as required to complete the proof in this case.

Case 2. α is a limit ordinal.

In this case, $I_\alpha = {}_{n<\alpha} I_n$ and $I_n \quad J$ for all $n \quad \alpha$ by hypothesis. Therefore, $I_\alpha \quad J$ as required.

Thus, the result follows by transfinite induction. ∎

The following definition is due to [Prz88]. Indeed it was shown in [Prz88] that each locally stratified program has a unique perfect model. Our proof in Theorem 8.1.9 below, using our previously obtained results, however, is more constructive.

8.1.8 Definition Suppose that P is a locally stratified normal logic program, and let l denote the associated level mapping. Given two distinct models M and N for P, we say that N is *preferable* to M if, for every ground atom A in $N \setminus M$, there is a ground atom B in $M \setminus N$ such that $l(A) > l(B)$. Finally, we say that a model M for P is *perfect* if there are no models for P preferable to M.

Notice that the requirement $l(A) > l(B)$ is dual to the requirement $A \quad B$ relative to the priority relation defined in [Prz88].

8.1.9 Theorem Suppose that P is a normal logic program which is locally stratified with respect to a level mapping $l : B_P \to \gamma$, where γ is a countable ordinal. Then $I_{[P]}$ is a perfect model for P and indeed is the only perfect model for P.

Proof: Suppose that there is a model N for P which is preferable to $I_{[P]}$ (and therefore distinct from $I_{[P]}$); we will derive a contradiction.

First note that $N \setminus I_{[P]}$ must be non-empty, otherwise we have $N \quad I_{[P]}$. But this inclusion forces equality of N and $I_{[P]}$ since $I_{[P]}$ is a minimal model for P, and therefore N and $I_{[P]}$ are not distinct. This means that there is a ground atom A in $N \setminus I_{[P]}$, which can be chosen so that $l(A)$ has minimum value; let B be a ground atom in $I_{[P]} \setminus N$ corresponding to A in accordance with Definition 8.1.8, and which satisfies $l(A) > l(B)$.

Next we note that $T_{[1]}(N) \quad T_P(N) \quad N$, since N is a model for P. Hence, N is a model for $P_{[1]}$, which implies that $I_1 \quad N$ since I_1 is the least model for the definite program $P_{[1]}$. Therefore, B can be chosen so that $B \in I_{n_0} \setminus N$, with minimal $n \; > 1$. Now n cannot be a limit ordinal, otherwise we would have $I_{n_0} = \quad {}_{m<n_0} I_m$, from which we would conclude that $B \in I_m \setminus N$ for some $m \quad n$ contrary to the choice of n . Thus, n must be a successor ordinal and, therefore, B can be chosen so that $B \in I_{[n_0,m_0]} \setminus N$, where m is such that $I_{[n_0,m_1]} \setminus N = \emptyset$ whenever m_1 , ; indeed, since $I_1 \quad N$, we must have $n \; > 1$ and $m \; \geq 1$ also. Consequently, $B \in T_{[n_0]}(I_{[n_0,m_0-1]}) \setminus N$ showing that there is a clause $B \leftarrow C_1, \ldots, C_{k_1}, \neg D_1, \ldots, \neg D_{l_1}$ in $P_{[n_0]}$ with the property that each $C_i \in I_{[n_0,m_0-1]}$ and no $D_j \in I_{[n_0,m_0-1]}$. Since $l(D_j) \quad -1$ for each j, we see that none of the D_j belong to $I_{[P]}$ by Lemma 8.1.5 (d). But all the C_i, if there are any, must belong to N by the choice of the numbers n and m . Moreover, there must be at least one D_j and indeed at least one belonging to N. For if there were no D_j or we had each $D_j \in N$, then we would have $B \in T_{P_{n_0}}(N) \quad T_P(N) \quad N$, using again the fact that N is a model for P. But this leads to the conclusion that $B \in N$, which is contrary to $B \in I_{[P]} \setminus N$. Thus, there is a $D = D_j \in N \setminus I_{[P]}$, for some j, satisfying $l(D) \quad (B) \quad (A)$. Since A was chosen in $N \setminus I_{[P]}$ to have smallest level, we have a contradiction.

This contradiction shows that $I_{[P]}$ must be a perfect model for P as required. The last statement in the theorem concerning uniqueness of $I_{[P]}$ now follows from [Prz88, Theorem 4]. ∎

Since it is shown in [Prz88] that perfect models are independent of the local stratification, we also have the following result.

8.1.10 Corollary If P is a normal logic program which is locally stratified with respect to two level mappings l_1 and l_2, then the corresponding models $I_{[P_1]}$ and $I_{[P_2]}$ are equal.

8.1.11 Program Since locally stratified programs are a generalization of locally hierarchical programs it is clear that each locally hierarchical program has a unique perfect model. This does not hold, however, for Φ^*-accessible programs. Indeed, the program

$$p \leftarrow \neg q$$
$$q \leftarrow r, \neg p$$

is Φ^*-accessible (even acceptable) with respect to the unique supported model $M = \{p\}$. However, $I = \{q\}$ is also a model of this program and while I is preferable to M, M in turn is also preferable to I, so P does not have a perfect model.

It also follows from [Prz88, Theorem 4] and Theorem 8.1.9 above that $I_{[P]}$ coincides with the model M_P of [ABW88] when P is stratified. However, for the sake of completeness we next present a proof of this fact using the methods established thus far. To do this, it will be convenient to introduce the concept $T \Uparrow n(I)$ for a mapping $T : I_P \to I_P$ and $I \in I_P$. In fact, $T \Uparrow n(I)$ is defined inductively as follows:

$$\begin{aligned} T \Uparrow 0(I) &= I \\ T \Uparrow (n \quad 1)(I) &= T(T \Uparrow n(I)) \quad I \\ T \Uparrow \omega(I) &= \quad \overset{\infty}{\underset{n=0}{}} T \Uparrow n(I). \end{aligned}$$

8.1.12 Theorem Let P be a stratified normal logic program. Then $I_{[P]} = M_P$.

Proof: As usual, we take the stratification to be $P = P_1 \quad \ldots \quad P_m$ and we will show by induction that $I_k = M_k$ for $k = 1, \ldots, m$ and that $I_k = M_m$ for $k > m$. From this we clearly have $I_{[P]} = M_m = M_P$ as required.

With the definition of the level mapping we are currently using and with the conventions we have made regarding the stratification, we note first that the equalities $P_{[k]} = \text{ground}(P_1 \quad P_2 \quad \ldots \quad P_k)$ and $P(k-1) = \text{ground}(P_k)$ both hold for $k = 1, \ldots, m$, where $P(k)$ is as defined in the proof of Lemma 8.1.5.

Now $P_{[1]} = \text{ground}(P_1)$ is definite, even if empty, and so it is immediate that $T_{P_1} \Uparrow i(M \ ) = T_{P_1} \uparrow i(M \ )$ for all i and that $I_1 = M_1$. So suppose next that $T_{P_{k+1}} \Uparrow i(M_k) = T_{P_{k+1}} \uparrow i(M_k)$ for all i and that $I_{k+1} = M_{k+1}$ for some $k > 0$. Then $T_{P_{k+2}} \Uparrow 0(M_{k+1}) = M_{k+1} = T_{P_{k+2}} \uparrow 0(M_{k+1})$ and also $I_{[k+2,0]} = I_{k+1} = M_{k+1} = T_{P_{k+2}} \uparrow 0(M_{k+1})$. So now suppose that $T_{P_{k+2}} \Uparrow m(M_{k+1}) = T_{P_{k+2}} \uparrow m(M_{k+1})$ and that $I_{[k+2,m]} = T_{P_{k+2}} \uparrow m(M_{k+1})$ for some $m > 0$. Then $T_{P_{k+2}} \Uparrow (m \quad)(M_{k+1}) = T_{P_{k+2}}(T_{P_{k+2}} \Uparrow m(M_{k+1})) \quad M_{k+1}$ and $T_{P_{k+2}} \uparrow (m \quad 1)(M_{k+1}) = T_{P_{k+2}}(T_{P_{k+2}} \uparrow m(M_{k+1})) \quad T_{P_{k+2}} \uparrow m(M_{k+1})$, and it is clear that $T_{P_{k+2}} \Uparrow (m \quad 1)(M_{k+1}) \quad T_{P_{k+2}} \uparrow (m \quad 1)(M_{k+1})$. For the reverse inclusion, we note that under our present hypotheses we have $T_{P_{k+2}} \uparrow (m \quad 1)(M_{k+1}) = T_{P_{k+2}}(T_{P_{k+2}} \Uparrow m(M_{k+1})) \quad T_{P_{k+2}} \Uparrow m(M_{k+1})$ and so it suffices to show that $T_{P_{k+2}} \Uparrow m(M_{k+1}) \quad T_{P_{k+2}}(T_{P_{k+2}} \Uparrow m(M_{k+1})) \quad M_{k+1}$ or in other words that $I_{[k+2,m]} \quad T_{P(k+1)}(I_{[k+2,m]}) \quad I_{k+1}$. Since this latter set is equal to $I_{[k+2,m+1]}$ by the recursion equations of Corollary 8.1.6, the inclusion we want follows from the monotonicity of the sets $I_{[k+2,m]}$ relative to m. We conclude, therefore, that $T_{P_{k+2}} \Uparrow (m \quad 1)(M_{k+1}) = T_{P_{k+2}} \uparrow (m \quad)(M_{k+1})$.

Finally, $I_{[k+2,m+1]} = I_{k+1} \quad T_{P(k+1)}(I_{[k+2,m]}) = M_{k+1} \quad T_{P_{k+2}}(T_{P_{k+2}} \uparrow m(M_{k+1})) = M_{k+1} \quad T_{P_{k+2}}(T_{P_{k+2}} \Uparrow m(M_{k+1})) = T_{P_{k+2}} \Uparrow (m \quad 1)(M_{k+1}) = T_{P_{k+2}} \uparrow (m \quad 1)(M_{k+1})$, by the conclusions of the previous paragraph. Therefore, $I_{[k+2,m+1]} = T_{P_{k+2}} \uparrow (m \quad 1)(M_{k+1})$. From this we obtain, by induction, the equality $I_{[k+2,m]} = T_{P_{k+2}} \uparrow m(M_{k+1})$ for all m and with it the equality $I_{k+2} = M_{k+2}$ as required. ■

The details of the induction proof just given also establish the following proposition.

8.1.13 Proposition Let $P = P_1 \quad \ldots \quad P_m$ be a stratified normal logic program. Then we have $T_{P_{k+1}} \Uparrow i(M_k) = T_{P_{k+1}} \uparrow i(M_k)$ for all i and $k = 0, \ldots, m-1$.

8.2 Weakly Perfect Model Semantics

When studying various classes of programs, the question naturally arises as to how such classes relate to other classes known in the literature. From the definition, it follows immediately that the unique supported model class of all locally hierarchical programs is contained in the class of all locally stratified programs. In this section, we will relate the class of all Φ-accessible programs to the notion of weak stratification.

It was pointed out in [BF91, Remark 5.3] that the original definition of weakly stratified programs in [PP90] is ambiguous since the two conditions

(a) All strata of a program P consist of trivial components only.

(b) All layers of a program P are definite programs.

which were originally used for defining weakly stratified programs are not equivalent. We will call a program *weakly stratified-a* if condition (a) holds, and *weakly stratified-b* if condition (b) holds. For a discussion of this, see [BF91, Section 5], and we refer to the same publication for notation concerning weakly stratified programs.

In [PZ98], it was shown that each acceptable program [AP93] is weakly stratified-a. From [GRS91, Corollary 4.3], we immediately obtain that each Φ-accessible program has a total well-founded model, ie. is *effectively stratified* [BF91]. Again from [BF91, Proposition 5.4], we obtain that a program which is weakly stratified-b, is also effectively stratified.

It is easy to see that a program which is weakly stratified-b, is also weakly stratified-a. In the opposite direction, we have the following result.

8.2.1 Theorem If P is weakly stratified-a and if there does not exist a clause $A \leftarrow$ `body` in ground(P) with $\neg A$ occurring in `body`, then P is weakly stratified-b.

Proof: Since P is weakly stratified-a, all minimal components are trivial. Let $A \leftarrow$ `body` be a clause in the bottom layer. Without loss of generality assume that `body` contains some negative literal $\neg B$, ie.[1] B , with $A = B$ by assumption. Since the component containing A is trivial, we obtain $A > B$ and therefore we obtain a contradiction. ■

It is clear from the last result that a locally hierarchical program is weakly stratified-a if and only if it is weakly stratified-b. This does in fact also hold for locally stratified programs.

We will now generalize a result from [PZ98], that all acceptable programs are weakly stratified-a.

[1] “$<$” denotes the dependency relation taken from the dependency graph of P [PP90].

8.2.2 Theorem If P is Φ-accessible, then P is weakly stratified-a and the unique supported model M_P of P is also its weakly perfect-a model.

Proof: Let M_P be the unique supported model of P and let l be its canonical level mapping wrt. Φ. We can also assume without loss of generality that for each level α there exists some $A \in B_P$ with $l(A) = \alpha$.
(1) We first show that all components of the bottom stratum $S(P)$ of P are trivial. Assume that this is not the case, i.e. that there exists a minimal component $C \quad S(P)$ which is not trivial. Then there must be some $A \in C$ with $l(A)$ minimal, and some $A' \in C$ with $A = A'$. Note that $A \quad A'$ and $A' \quad A$ [BF91, Definition 5.1]. Let B be an arbitrary atom occurring in a ground clause with head A. Then $B \quad A'$ and therefore $B \quad A$, and by minimality of C we obtain $B \in C$. So all atoms B occurring in bodies of clauses in ground(P) with head A belong to C. Since P is Φ-accessible, however, there must exist some choice of B for which we have $l(B) \quad (A)$, and this contradicts the minimality of $l(A)$. Note $<l$
that the bottom stratum contains all atoms of level 0, and hence is non-empty.
(2) The model M of the bottom layer is compatible with M_P, i.e. if a literal is true, respectively false, in M, then it is true, respectively false, in M_P. In order to see this, note that for every atom A in a minimal component, the bottom layer $L(P)$ contains all clauses with head A and all clauses with head being any of the body atoms of clauses in the bottom layer. Since the program P is Φ-accessible, it is easy to see that the subprogram formed by the bottom layer is also Φ-accessible and has a unique supported model which is compatible with M_P.
Now let A be an atom in $L(P)$ which occurs negatively in the body of some clause. Since all components are trivial, A must also be the head of the same clause, i.e we have $A \quad A$. If B is another body atom in the same clause, then we obtain $B \qquad$ and $A \quad B$ which contradicts triviality of all components. Hence, $<A$
if some atom A occurs negatively in a clause in $L(P)$, then the clause is of the form $A \leftarrow \neg A$. All models of $L(P)$ must therefore assign the truth value true to all atoms occurring negatively in $L(P)$. The program which is obtained from omitting all these clauses is definite and has a least model which agrees with M_P. If we add to this model all atoms which occur negatively in $L(P)$, we obtain the least model of $L(P)$.
(3) We show that P/M is Φ-accessible (see [BF91]). This is indeed the case since (2) holds, and is easily seen by applying Theorem 6.5.3.
(4) We can now apply steps (1), (2) and (3) via transfinite induction as in [PP90], which yields that P is indeed weakly stratified-a and that M_P is the weakly perfect-a model of P. Thus, the proof is complete. ■

8.2.3 Theorem Let P be Φ-accessible. Then P has a unique supported model M_P which is the unique stable model, the well-founded model, a minimal two-valued model, and the weakly perfect-a model of P.

Proof: We know that $M_P = \Phi_P \uparrow \alpha$ for some ordinal α and that M_P is total. By Theorem 6.5.4, we know that M_P^+ is a minimal two-valued model of P, and by

Theorem 8.2.2 we know that M_P is the weakly perfect-a model of P. By [GRS91, Corollary 4.3], $M_P = \Phi_P\!\uparrow\!\alpha$ is a subset of the well-founded model of P, and since M_P is total, it must coincide with the well-founded model. By [GRS91, Corollary 5.6], totality of the well-founded model implies that it coincides with the unique stable model of the program. This completes the proof. ■

8.2.4 Program Acceptable programs are not necessarily weakly stratified-b, as can be seen from the following program.

$$p \leftarrow$$
$$p \leftarrow q, \neg p$$

The bottom layer contains the clause $p \leftarrow q, \neg p$ and is therefore not a definite program.

8.2.5 Program On the other hand, there exist programs with unique supported models which are not weakly stratified-a. To see this, note that the following program

$$p \leftarrow \neg q$$
$$q \leftarrow \neg p$$
$$p \leftarrow \neg p$$

has unique supported model $\{p\}$. However, it has $\{p, q\}$ as a minimal component which is not trivial.

8.3 Summary and Further Work

We have provided an iterative approach to the perfect model semantics of locally stratified programs and located the classes of programs discussed in Chapters 5 and 6 in the context of other standard semantics. Figure 8.1 on page 134 extends Figure 5.1 on page 91 incorporating the results from Section 8.2.

Of course, the results in Section 8.1 indicate possible research concerning the extent to which iterative approaches can be applied to other semantics. The results in Section 8.2 clarify some relationships between classes of programs known from the literature, which also is a field of further study.

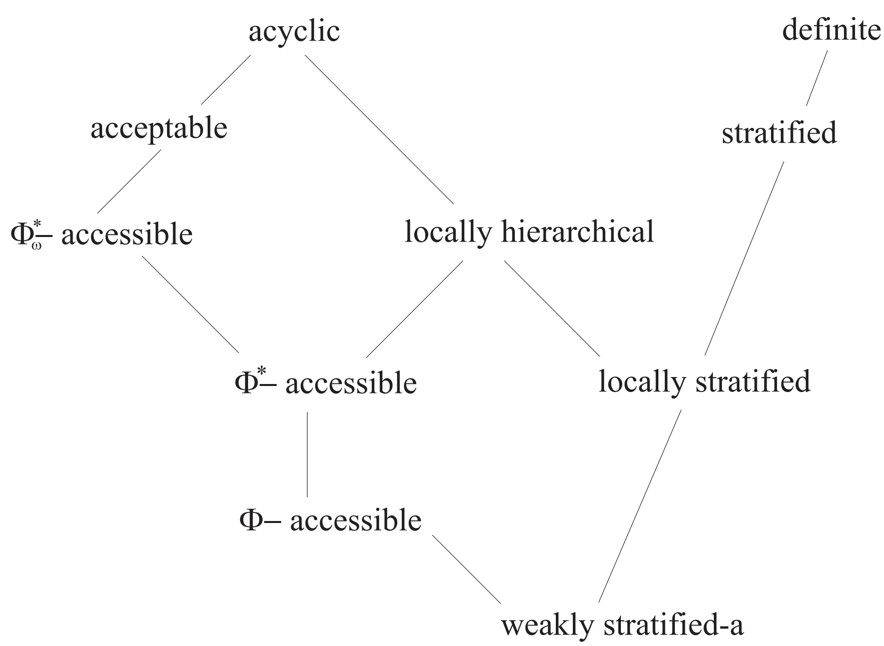

Figure 8.1: Dependencies between classes of programs. If a class is depicted lower in the diagram, this indicates that it is more general.

Chapter 9

Logic Programs and Neural Networks

Logic Programs and Neural Networks are two important paradigms in Artificial Intelligence. Their abilities, and our theoretical understanding of them, however, seem to be rather complementary. Logic Programs are highly recursive and well understood from the point of view of declarative semantics. Neural Networks can be trained but yet lack a declarative reading. Recent publications, for example [BDJ^{+}99, HK94, HSK99, Zha99], suggest studying the relationships between the two paradigms with the long-term aim of merging them in such a way that the advantages of both can be combined.

The results we wish to discuss draw heavily on the work of Holldobler, Kalinke and Störr [HK94, HSK99], which we will in part generalize. It will be convenient to briefly review their approach and their results. For our investigations, it will be sufficient to consider Herbrand interpretations only.

In [HK94], a strong relationship between propositional logic programs, i.e. programs without variable or function symbols, and 3-layer feedforward and recurrent networks was established. For each such program P, a 3-layer feedforward network can be constructed which computes the single-step operator T_P associated with P. To this end, each atom in P is represented by one or more units in the network. If the program is such that iterates of T_P, for any initial value, converge to a unique fixed point of T_P, then the network can be cast into a recurrent network which settles down into a unique stable state corresponding to the fixed point. On the other hand, for each 3-layer network a propositional logic program P can be constructed such that the corresponding operator T_P is computed by the network.

In [HSK99], an attempt was made to obtain similar results for logic programs which are not propositional, that is, for programs which allow variables. The main obstacle which has to be overcome in this case is that the Herbrand base is in general infinite; it is therefore not possible to represent an atom by one or more units in the network. The solution suggested in [HSK99] uses a general result due to Funahashi [Fun89], see Theorem 9.1.1, which states that every continuous function on a compact subset of the real numbers can be uniformly approximated by certain types of 3-layer neural networks. By casting the T_P-operator into such a function, approximating the single-step operator is shown to be possible.

In order to obtain a continuous real-valued function from T_P, metrics were employed in [HSK99]. For acyclic[1] logic programs, a complete metric can be obtained which renders the single-step operator a contraction, see Section 5.1. By identifying the single-step operator with a mapping on the reals, a contractive, and therefore continuous, real-valued function is obtained which represents the single-step operator. This function can in turn be approximated by neural networks due to the result of Funahashi mentioned above. For certain kinds of acyclic programs, namely such which admit an injective level mapping, the resulting network can then again be cast into a recurrent network which settles down into a unique stable state corresponding to the unique fixed point of the operator.

In this chapter, we will investigate a more general approach to representing the single-step operator for (non-propositional) normal logic programs by neural networks.

In Section 9.1, we will use Theorem 4.2.6 which characterizes continuity of the single-step operator in the atomic topology, and apply the approximation theorem of Funahashi in order to approximate single-step operators by neural networks.

In Section 9.2, we will show that for any given normal logic program, its associated single-step operator can be realized as a Borel-measurable real-valued function. An approximation theorem due to Hornik, Stinchcombe and White [HSW89], see Theorem 9.2.1, can then be applied to show that each single-step operator for any normal logic program can be approximated arbitrarily well by neural networks in a metric ϱ_μ defined in measure-theoretic terms in Section 9.2.

Cantor Topology

Recall from Section 4.2, that I_P can be identified with the powerset of B_P, and that it can therefore also be identified with the set 2^{B_P} of all functions from B_P to $\{0,1\}$ (or to any other two-point space). Using this latter identification, the topology Q becomes a topology on the function space 2^{B_P}, and is exactly the product topology (of point-wise convergence) on 2^{B_P} if the two-point space is endowed with the discrete topology.

If we interpret I_P as the set of all functions from B_P to $\{0,2\}$, so that we now take the two-point space as $\{0,2\}$, we can identify I_P with the set of all those real numbers in the unit interval $[0,1]$ which can be written in ternary form without using the digit 1; in other words we can identify I_P with the *Cantor set*. The product topology mentioned above then coincides with the subspace topology inherited from the natural topology on the real numbers, and the resulting space is called the *Cantor space* $\mathcal{C}$. Thus, the Cantor space $\mathcal{C}$ is homeomorphic

[1] These programs were called *recurrent* in [HSK99].

to the topological space (I_P, Q), and in the following $\iota : I_P \to \mathcal{C}$ will denote a homeomorphism between I_P and $\mathcal{C}$. It is well-known that the Cantor space is a compact subset of $\mathbb{R}$, and we can define $l(x) = \max\{y \in \mathcal{C} : y \leq x\}$ and $u(x) = \min\{y \in \mathcal{C} : y \geq x\}$ for each $x \in [0,1]$.

Neural Networks

A *3-layer feedforward network* (or *single hidden layer feedforward network*) consists of an *input layer*, a *hidden layer*, and an *output layer*. Each layer consists of finitely many *computational units*. There are connections from units in the input layer to units in the hidden layer, and from units in the hidden layer to units in the output layer. The input-output relationship of each unit is represented by *inputs* x_i, *output* y, *connection weights* w_i, *threshold* θ, and a function ϕ as follows:

$$y = \phi\left(\sum_i w_i x_i - \theta\right).$$

The function ϕ, which we will call the *squashing function* of the network, is usually non-constant, bounded and monotone increasing, and sometimes also assumed to be continuous. We will specify the requirements on ϕ that we assume in each case.

We assume throughout that the input-output relationships of the units in the input and output layer are linear. The output function of a network as described above is then obtained as a mapping $f : \mathbb{R}^r \to \mathbb{R}$ with

$$f(x_1, \ldots, x_r) = \sum_j c_j \phi\left(\sum_i w_{ji} x_i - \theta_j\right),$$

where r is the number of units in the input layer and the constants c_j correspond to weights from hidden to output layers.

We refer to [Bis95] for background concerning artificial neural networks.

Measurable Functions

A collection M of subsets of a set X is called a *σ-algebra* if (i) $\emptyset \in M$; (ii) if $A \in M$ then its complement ${}^cA \in M$; (iii) if (A_n) is a sequence of sets in M, then the union $\bigcup A_n \in M$. The pair (X, M) is called a *measurable space*. A function $f : X \to X$ is said to be *measurable* with respect to M if $f^{-1}(A) \in M$ for each $A \in M$.

If M is a collection of subsets of a set X, then the smallest σ-algebra $\sigma(M)$ containing M is called the σ-algebra *generated by* M. In this case, a function $f : X \to X$ is measurable with respect to $\sigma(M)$ if and only if $f^{-1}(A) \in \sigma(M)$ for each $A \in M$. If $\mathcal{B}$ is the subbase of a topology $\mathcal{T}$, and $\mathcal{B}$ is countable, then $\sigma(\mathcal{B}) = \sigma(\mathcal{T})$. If $\mathcal{B}$ is a subbase of the natural topology on $\mathbb{R}$, then $\sigma(\mathcal{B})$ is called the *Borel-σ-algebra* on $\mathbb{R}$, and a function which is measurable with respect to this σ-algebra is called *Borel-measurable*. A measure on $(\mathbb{R}, \sigma(\mathcal{B}))$ is called a *Borel-measure*.

We refer the reader to [Bar66, Bau92] for background concerning elementary measure theory.

9.1 Approximating Continuous Single-Step Operators by Neural Networks

Under certain conditions, given in Theorem 4.2.6, the single-step operator associated with a logic program is continuous in the atomic topology. By identifying the space of all interpretations with the Cantor space, a continuous function on the reals is obtained which can be approximated by 3-layer feedforward networks. We investigate this next.

The following Theorem can be found in [Fun89, Theorem 2].

9.1.1 Theorem Suppose that $\phi : \mathbb{R} \to \mathbb{R}$ is non-constant, bounded, monotone increasing and continuous. Let $\quad \mathbb{R}^n$ be compact, let $f : \quad \to \mathbb{R}$ be a continuous mapping and let $\varepsilon > 0$. Then there exists a 3-layer feedforward network with squashing function ϕ whose input-output mapping $\bar{f} : \quad \to \mathbb{R}$ satisfies $\max_{x \in K} d(f(x), \bar{f}(x)) \quad \varepsilon$, where d is a metric which induces the natural topology on $\mathbb{R}$.

In other words, each continous function $f : \quad \to \mathbb{R}$ can be uniformly approximated by input-output functions of 3-layer networks.

We already know that the Cantor space $\mathcal{C}$ is a compact subset of the real line and that the topology which $\mathcal{C}$ inherits as a subspace of $\mathbb{R}$ coincides with the Cantor topology on $\mathcal{C}$. Also, the Cantor space $\mathcal{C}$ is homoeomorphic to I_P endowed with the atomic topology Q, see Theorem 4.2.4. Hence, if the T_P-operator is continuous in Q, we can identify it with a mapping $\iota(T_P) : \mathcal{C} \to \mathcal{C} : x \to \iota(T_P(\iota^{-1}(x)))$ which is continous in the subspace topology of $\mathcal{C}$ in $\mathbb{R}$.

9.1.2 Theorem Let P be a normal logic program. If, for each $I \in I_P$ and for each $A \in B_P$ with $A \in T_P(I)$, either there is no clause in P with head A or there is a finite set $S(I, A) = \{A_1, \ldots, A_k, B_1, \ldots, B_k\}$ of elements of B_P satisfying the properties (i) and (ii) of Theorem 4.2.6, then T_P (more precisely $\iota(T_P)$) can be uniformly approximated by input-output mappings of 3-layer feedforward networks.

In particular, this holds for the operator T_P if P does not contain any local variables or is acyclic with injective level mapping.

Proof: Under the conditions stated in the theorem, the single-step operator T_P is continuous in the atomic topology. Using a homeomorphism $\iota : I_P \to \mathcal{C}$, the resulting function $\iota(T_P)$ is continuous on the Cantor space $\mathcal{C}$, which is a compact subset of $\mathbb{R}$. Applying Theorem 9.1.1, $\iota(T_P)$ can be uniformly approximated by input-output functions of 3-layer feedforward networks.

Now if P does not contain any local variables, then T_P is obviously continuous in Q by Theorem 4.2.6. Now let P be acyclic with injective level mapping and let

$A \in B_P \setminus T_P(I)$ for some $I \in I_P$. Since the level mapping is finite, there exist only finitely many atoms which occur in bodies of clauses with head A, which suffices by Theorem 4.2.6. ■

9.2 Approximating the Single-Step Operator by Neural Networks

By Theorem 9.1.1, continuous functions can be uniformly approximated by input-output functions of 3-layer feedforward networks. It is also possible to approximate each measurable function on $\mathbb{R}$, but in a much weaker sense. We will investigate this in the present section.

The following was given in [HSW89, Theorem 2.4]

9.2.1 Theorem Suppose that ϕ is a monotone increasing function from $\mathbb{R}$ onto $(0,1)$. Let $f : \mathbb{R}^r \to \mathbb{R}$ be a Borel-measurable function and let μ be a probability Borel-measure on $\mathbb{R}^r$. Then, given any $\varepsilon > 0$, there exists a 3-layer feedforward network with squashing function ϕ whose input-output function $\bar{f} : \mathbb{R}^r \to \mathbb{R}$ satisfies

$$\varrho_\mu(f, \bar{f}) = \inf\{\delta > 0 : \mu\{x : |f(x) - \bar{f}(x)| > \delta\} \qquad \} \qquad .$$

In other words, the class of functions computed by 3-layer feedforward neural nets is dense in the set of all Borel-measurable functions $f : \mathbb{R}^r \to \mathbb{R}$ relative to the metric ϱ_μ defined in Theorem 9.2.1.

We have already noted that the operator T_P is not continuous in the topology Q in general, nor is it continuous in the Scott topology on I_P in general. We proceed to show next that the single step operator has the pleasing property that it is measurable with respect to $\sigma(Q)$ for arbitrary programs, and therefore that it can always be extended to a Borel-measurable function on $\mathbb{R}$.

9.2.2 Proposition Let P be a normal logic program and let T_P be its associated single-step operator. Then T_P is measurable on $(I_P, \sigma(\mathcal{G})) = (I_P, \sigma(Q))$.

Proof: We need to show that for each subbasic set $\mathcal{G}(L)$, we have $T_P^{-1}(\mathcal{G}(L)) \in \sigma(\mathcal{G})$.

First, let $L = A$ be an atom. If A is not the head of any clause in ground(P), then $T_P^{-1}(\mathcal{G}(A)) = \emptyset \in \sigma(\mathcal{G})$. If A is the head of a clause in ground(P), then there are at most countably many clauses

$$A \leftarrow A_{i1}, \dots, A_{ik_i}, \neg B_{i_1}, \dots, \neg B_{il_i}$$

in ground(P) with head A, and we obtain

$$T_P^{-1}(\mathcal{G}(A)) = \underset{i}{} \mathcal{G}(A_{i1}, \dots, A_{ik_i}, \neg B_{i_1}, \dots, \neg B_{il_i})$$

which is indeed in $\sigma(\mathcal{G})$.

Now suppose that $L = \neg A$ is a negative literal. If A is not the head of any clause in ground(P), then $T_P^{-1}(\mathcal{G}(\neg A)) = I_P \in \sigma(\mathcal{G})$. So assume that A is the head of some clause in ground(P). If there is a unit clause with head A, then $T_P^{-1}(\mathcal{G}(\neg A)) = \emptyset \in \sigma(\mathcal{G})$. So assume that none of the clauses in ground(P) with head A is a unit clause. Then there are at most countably many clauses

$$A \leftarrow A_{i1}, \dots, A_{ik_i}, \neg B_{i_1}, \dots, \neg B_{il_i}$$

in ground(P) with head A. We then obtain

$$T_P^{-1}(\mathcal{G}(\neg A)) = \mathop{}\limits_{i} \mathcal{G}(\neg A_{i1}) \qquad \mathcal{G}(\neg A_{ik_i}) \quad (B_{i_1}) \qquad (B_{il_i})$$

which is indeed in $\sigma(\mathcal{G})$. ∎

By means of Proposition 9.2.2, we can now view the operator T_P as a measurable function $\iota(T_P)$ on $\mathcal{C}$ by identifying I_P with $\mathcal{C}$ via the homeomorphism ι. Since $\mathcal{C}$ is measurable as a subset of the real line, this operator can be extended[2] to a measurable function on $\mathbb{R}$ and we obtain the following result.

9.2.3 Theorem Given any normal logic program P, the associated operator T_P (more precisely $\iota(T_P)$) can be approximated in the manner of Theorem 9.2.1 by input-output mappings of 3-layer feedforward networks.

This result is somewhat unfortunate since the approximation stated in Theorem 9.2.1 is only *almost everywhere*, i.e. pointwise with the exception of a set of measure zero. The Cantor set, however, is a set of measure zero. Nevertheless, we are able to strengthen this result a bit by giving an explicit extension of T_P to the real line. We define a sequence (T_n) of measurable functions on $\mathbb{R}$ as follows, where $l(x)$ and $u(x)$ are as defined earlier, and for each $i \in \mathbb{N}$ we set

$$D_i = \mathop{}\limits_{k=1}^{\frac{3^i-1}{2}} [(2k-1)3^{-i}, 2k \quad 3^{-i}],$$

and for each $i \geq 2$ we define

$$T\ (x) = \begin{cases} \iota(T_P)(x) & \text{if } x \in \mathcal{C} \\ \iota(T_P)(0) & \text{if } x < \ 0 \\ \iota(T_P)(1) & \text{if } x > 1 \\ 0 & \text{otherwise} \end{cases}$$

$$T_1(x) = \begin{matrix} \iota(T_P)(l(x)) \quad \frac{\iota(T_P)(u(x)) - \iota(T_P)(l(x))}{u(x) - l(x)}(x - l(x)) & \text{if } x \in D_1 \\ 0 & \text{otherwise} \end{matrix}$$

$$T_i(x) = \begin{matrix} \iota(T_P)(l(x)) \quad \frac{\iota(T_P)(u(x)) - \iota(T_P)(l(x))}{u(x) - l(x)}(x - l(x)) & \text{if } x \in D_i \\ 0 & \text{otherwise.} \end{matrix}$$

[2]E.g. as a function $T : \mathbb{R} \to \mathbb{R}$ with $T(x) = \iota(T_P(\iota^{-1}(x)))$ if $x \in \mathcal{C}$ and $T(x) = 0$ otherwise.

We define the function $T : \mathbb{R} \to \mathbb{R}$ by $T(x) = \sup_i T_i(x)$ and obtain $T(x) = \iota(T_P(x))$ for all $x \in \mathcal{C}$ and $T(\iota(I)) = \iota(T_P(I))$ for all $I \in I_P$. Since all the functions T_i, for $i \geq 1$, are piecewise linear and therefore measurable, the function T is also measurable. Intuitively, T is obtained by a kind of linear interpolation.

If $i : B_P \to \mathbb{N}$ is a bijective mapping, then we can obtain a homeomorphism $\iota : I_P \to \mathcal{C}$ from i as follows: we identify $I \in I_P$ with $x \in \mathcal{C}$ where x written in ternary form has 2 as its $i(A)$th digit (after the decimal point) if $A \in I$, and 0 as its $i(A)$th digit if $A \in I$. If $I \in I_P$ is finite or cofinite[3], then the sequence of digits of $\iota(I)$ in ternary form is eventually constant 0 (if I is finite) or eventually constant 2 (if I is cofinite). Thus, each such interpretation is the endpoint of a linear piece of one of the functions T_i, and therefore of T.

9.2.4 Corollary Given any normal logic program P, its single-step operator T_P (more precisely $\iota(T_P)$) can be approximated by input-output mappings of 3-layer feedforward networks in the following sense: for every $\varepsilon > 0$ and for every $I \in I_P$ which is either finite or cofinite, there exist a 3-layer feedforward network with input-output function f and $x \in [0, 1]$ with $|x - \iota(I)|$ such that $|\iota(T_P(I)) - f(x)|$. $<\varepsilon$

Proof: We use a homeomorphism ι which is obtained from a bijective mapping $i : B_P \to \mathbb{N}$ as in the paragraph preceeding the Corollary. We can assume that the measure μ from Theorem 9.2.1 has the property that $\mu\{[x, x \quad \varepsilon]\} \leq \varepsilon$ for each $x \in \mathbb{R}$. Let $\varepsilon > 0$ and $I \in I_P$ be finite or cofinite. Then by construction of T there exists an interval $[\iota(I), \iota(I)+ \ \delta]$ with $\delta \quad \frac{\varepsilon}{2}$ (or analogously $[\iota(I) - \delta, \iota(I)]$) such that T is linear on $[\iota(I), \iota(I)+\delta]$ and $|T(\iota(I))-T(x)| \quad \frac{\varepsilon}{2}$ for all $x \in [\iota(I), \iota(I)+\delta]$. By Theorem 9.2.1 and the previous paragraph, there exists a 3-layer feedforward network with input-output function f such that $\varrho_\mu(T, f) \quad \delta$, that is, $\mu\{x : |T(x) - f(x)| > \delta\}$. By our condition on μ, there is $x \in [\iota(I), \iota(I)+ \ \delta]$ with $<$ $|T(x)-f(x)| \leq \delta< \ \frac{\varepsilon}{2}$. We can conclude that $|\iota(T_P(I))-f(x)| = |T(\iota(I))-f(x)| \leq |T(\iota(I)) - T(x)| \quad |T(x) - f(x)|$ as required. ■

It would be of interest to strengthen this approximation for sets other than the finite and cofinite elements of I_P, although it is interesting to note that the finite interpretations correspond to compact elements in the sense of domain theory, see [SHLG94] and Definition 1.1.4.

9.3 Summary and Further Work

There are two aspects to this work. On the one hand, one can consider the problem of approximating the T_P operator, associated with logic programs P, by means of input-output functions of multi-layer neural networks, as we have done here. This, in detail, involves relating properties of the network to classes of programs for which the approximation is possible. It also involves the consideration of what

[3] $I \quad I_P$ is *cofinite* if $B_P \setminus I$ is finite.

mathematical notions of approximation are useful and appropriate. Here we have discussed two well-known ones: uniform approximation on compacta, and a notion of approximation closely related to convergence in measure. Both these strands need further investigation, and this section is an account of work to date which is at an early stage of development. In the other direction, and we have not discussed this at all here except in passing, is to view logic programs as fundamental and to view the approximation process as a means of giving semantics to neural networks based on the declarative semantics of logic programs. There is considerable point in doing this in that the semantics of logic programming is well understood whilst that of neural networks is not, but is something to be taken up elsewhere, probably including work on quantitative logic programming as in [Mat99].

At the detailed mathematical level, the mapping $P \to T_P$ is not injective. So, although the single-step operator can basically be used to represent a program semantically, different programs may have the same single-step operator. This fine tuning is lost by our representation of logic programs by neural networks. However, passing to classes of programs with the same single-step operator is something that is often done in the literature on semantics and in fact is exactly the notion of *subsumption equivalence* due to [Mah88]. Moreover, there exist uncountably many homeomorphisms $\iota : I_P \to \mathcal{C}$; for example, every bijective mapping from B_P to $\mathbb{N}$ gives rise to such a homeomorphism as observed in the paragraph preceeding Corollary 9.2.4. So there is a lot of flexibility in the choice of ι and therefore in how one embeds I_P in $\mathbb{R}$. The homeomorphism used in [HSK99] employed the quaternary number system.

In [HSK99], as mentioned in the beginning of this chapter, the neural network obtained by applying the approximation theorem of Funahashi was cast into a recurrent network which settled down in a unique stable state corresponding to the unique fixed point of the single-step operator of the underlying program P. Strong assumptions had to be placed on P to make this possible: P was required to be acyclic with an injective level mapping. Acyclicity of the program yields the existence of a complete metric on I_P with respect to which its single-step operator is a contraction, see Section 5.1. For larger classes of programs, such as the Φ^*-accessible programs, we have seen that it is also possible to find metrics such that the single-step operator is a contraction: In Section 5.3 we have seen how to construct a complete d-metric ϱ for a given Φ^*-accessible program P, and since T_P is a contraction with respect to ϱ, see Proposition 5.3.4, it is also a contraction with respect to the complete metric d associated with ϱ as in Proposition 3.1.11.

It turns out, however, that the metric d thus obtained cannot in general be topologically imbedded into the real line. In order to see this, note that for the d-metric ϱ associated with a Φ^*-accessible program there may be an uncountable number of interpretations such that $\varrho($ $K) = 0$, namely for example all with I. Each such , however, becomes an isolated point with respect to the topology induced by d, i.e. the singleton set containing is open and closed in this topology. Now, if (I_P, d) could be topologically imbedded in the real line using an imbedding ι, then for each as above we would have that

$\{\iota(\quad)\}$ is open and closed in the topological subspace $\iota(I_P)$ of the real line, i.e. that there is an open interval $J \subset \mathbb{R}$ such that $J \cap \{\iota(I_P)\} = \{\iota(\quad)\}$. Assuming uncountably many isolated points in (I_P, d), we could therefore construct a partition of $\mathbb{R}$ into uncountably many intervals, which is impossible by a well-known result from general topology. Hence we conclude that (I_P, d) cannot in general be topologically imbedded into the real line.

From the considerations just presented we conclude that alternative metrics or even methods have to be investigated in order to carry over the result from [HSK99] mentioned above for acyclic programs with injective level mappings to more general classes.

Chapter 10

Conclusions

There are many aspects to this work, which are in fact closely interconnected. We want to conclude with a short discussion of different points of view from which the work in this thesis can be put into a more general perspective.

Logic Programming and Non-monotonic Reasoning

The denotational aspects of logic programming with negation are still not sufficiently understood. We contribute to this general line of research by using topological methods for the analysis of fixed-point semantics. Recently, some studies of topological approaches to inductive logic programming have been undertaken [GNAJBD00] which is a field of further study.

Knowledge Representation and Reasoning

Logic programming can also be understood as a simple model of reasoning, and the behaviour of the single-step operator as an inductive perspective on it. Since many of our results were concerned with understanding the dynamics of this operator, they can be understood as an approach to understand the dynamics of reasoning, as motivated for example in [BDJ+99]. Extensions, e.g. to quantitative logic programming paradigms which incorporate probabilistic or fuzzy logic structures, suggest themselves.

Comparison and Integration of Paradigms

The single-step operator obtains its iterative behaviour from a relatively simple set of rules, has a very complex dynamics which is difficult to understand, and sometimes produces meaningful results as limits of the iterations. From this perspective, analogies to chaos theory and topological dynamical systems come into view, and indeed some few investigations along these lines have already been undertaken. They also open up connections to other paradigms like artificial neural networks, as in Chapter 9.

Denotational Semantics and Domain Theory

In recent years, quantitative aspects of domain theory, using generalized metrics, have been studied intensively. The study of denotational logic programming semantics from a generalized metric point of view can be understood as a contribution to this general area of research. It is not surprising, for example, that injectivity of level mappings has made its appearance in several chapters, since the finite and cofinite interpretations correspond to the notion of compact elements in domain theory.

Investigations concerning domain theory in logic programming have also been undertaken by Rounds and Zhang [ZR97a, ZR97b, RZ98, ZR98], and relationships between their approach and the results in this thesis remain to be worked out. The topological perspective of our work gives a continuous point of view on the discrete logic programming paradigm and should also be transferable to quantitative logic programming paradigms as mentioned above.

Topology (in Computer Science)

General topology allows one to naturally build a bridge between the discrete and the continuous, which is an important line to investigate since computing is inherently discrete while the world, which computing is supposed to model, is often perceived as continous. The results in this thesis contribute to this discussion by providing a continous framework for the study of the discrete logic programming paradigm, as it was also suggested in [BDJ$^+$99]. We have also contributed to some topological aspects of domain theory and to the study of fixed-point theorems in general.

The author hopes that his results constitute valuable contributions to the above mentioned areas of research.

Bibliography

[AB90] K.R. Apt and M. Bezem. Acyclic programs. In D.H.D. Warren and P. Szeredi, editors, *Proceedings of the Seventh International Conference on Logic Programming*, pages 617–633. MIT Press, Cambridge, MA, 1990.

[ABW88] K.R. Apt, H.A. Blair, and A. Walker. Towards a theory of declarative knowledge. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 89–148. Morgan Kaufmann, Los Altos, CA, 1988.

[And97] J.H. Andrews. A logical semantics for depth-first prolog with ground negation. *Theoretical Computer Science*, 184(1–2):105–143, 1997.

[AP93] K.R. Apt and D. Pedreschi. Reasoning about termination of pure prolog programs. *Information and Computation*, 106:109–157, 1993.

[AP94] K.R. Apt and D. Pedreschi. Modular termination proofs for logic and pure prolog programs. In G. Levi, editor, *Advances in Logic Programming Theory*, pages 183–229. Oxford University Press, 1994.

[Apt95] K.R. Apt. Program verification and prolog. In E. Börger, editor, *Specification and Validation Methods for Programming Languages and Systems*, pages 55–95. Oxford University Press, 1995.

[Bar66] R.G. Bartle. *The Elements of Integration.* John Wiley & Sons, New York, 1966.

[Bat89] A. Batarekh. *Topological Aspects of Logic Programming.* PhD thesis, Syracuse University, June 1989.

[Bau92] H. Bauer. *Maß- und Integrationstheorie.* De Gruyter, Berlin, 1992.

[BDJ+99] H.A. Blair, F. Dushin, D.W. Jakel, A.J. Rivera, and M. Sezgin. Continuous models of computation for logic programs. In K.R. Apt, V.W. Marek, M. Truszczyński, and D.S. Warren, editors, *The*

Logic Programming Paradigm: A 25 Year Perspective, pages 231–255. Springer, Berlin, 1999.

[Bez89] M. Bezem. Characterizing termination of logic programs with level mappings. In E.L. Lusk and R.A. Overbeek, editors, *Proceedings of the North American Conference on Logic Programming*, pages 69–80. MIt Press, Cambridge, MA, 1989.

[BF91] N. Bidoit and C. Froideveaux. Negation by default and unstratifiable logic programs. *Theoretical Computer Science*, 78:85–112, 1991.

[BFMS98] R. Barbuti, N. De Francesco, P. Mancarella, and A. Santone. Towards a logical semantics for pure prolog. *Science of Computer Programming*, 32(1–3):145–176, 1998.

[Bis95] C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[BMPC99] S. Bouamama, D. Misane, and S. Prieß-Crampe. An application of ultrametric spaces in logic programming. Preprint, 5 pages, July 1999.

[BS89a] A. Batarekh and V.S. Subrahmanian. The query topology in logic programming. In *Proc. 1989 Symposium on Theoretical Aspects of Computer Science*, volume 349 of *Lecture Notes in Computer Science*, pages 375–387. Springer, Berlin, 1989.

[BS89b] A. Batarekh and V.S. Subrahmanian. Topological model set deformations in logic programming. *Fundamenta Informaticae*, 12:357–400, 1989.

[BvBR96] M.M. Bonsangue, F. van Breugel, and J.J.M.M. Rutten. Alexandroff and Scott topologies for generalized metric spaces. In S. Andima et al., editor, *Papers on General Topology and Applications: Eleventh Summer Conference at University of Southern Maine*, Annals of the New York Academy of Sciences, pages 49–68, 1996.

[Cav89] L. Cavedon. Continuity, consistency, and completeness properties for logic programs. In G. Levi and M. Martelli, editors, *Proceedings of the 6th International Conference on Logic Programming*, pages 571–584. MIT Press, Cambridge, MA, 1989.

[Cav91] L. Cavedon. Acyclic programs and the completeness of SLDNF-resolution. *Theoretical Computer Science*, 86:81–92, 1991.

[Cha88] D. Chan. Constructive negation based on the completed database. In *Proc. of the 5th Int. Conf. and Symp. on Logic Programming*, pages 111–125, 1988.

[Cla78] K.L. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 293–322. Plenum Press, New York, 1978.

[CS00] E. Clifford and A.K. Seda. Uniqueness of the fixed-points of single-step operators in many-valued logics. *Journal of Electrical Engineering*, 51(12/s), 2000. Proceedings of the 2nd Slovakian Student Conference in Applied Mathematics (SCAM2000), Bratislava. Slovak Academy of Sciences. To appear.

[DG82] J. Dugundji and A. Granas. *Fixed Point Theory*. Monografie Matematyczne. Polish Scientific Publishers, Warsaw, 1982.

[EH98] A. Edalat and R. Heckmann. A computational model for metric spaces. *Theoretical Computer Science*, 193:53–73, 1998.

[Fag91] F. Fages. A new fixpoint semantics for general logic programs compared with the well-founded and the stable model semantics. *New Generation Computing*, 9:425–443, 1991.

[Fag94] F. Fages. Consistency of Clark's completion and existence of stable models. *Journal of Methods of Logic in Computer Science*, 1:51–60, 1994.

[FBJ90] M. Fitting and M. Ben-Jacob. Stratified, weak stratified, and three-valued semantics. *Fundamenta Informaticae*, XIII:19–33, 1990.

[Fer94] A.P. Ferry. *Topological Characterizations for Logic Programming Semantics*. PhD thesis, University of Michigan, 1994.

[Fit85] M. Fitting. A Kripke-Kleene-semantics for general logic programs. *Journal of Logic Programming*, 2:295–312, 1985.

[Fit94] M. Fitting. Metric methods: Three examples and a theorem. *Journal of Logic Programming*, 21(3):113–127, 1994.

[FL82] P. Fletcher and W.F. Lindgren. *Quasi-uniform Spaces*. Dekker, 1982.

[Fun89] K. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183–192, 1989.

[GL88] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In R.A. Kowalski and K.A. Bowen, editors, *Logic Programming. Proceedings of the 5th International Conference and*

Symposium on Logic Programming, pages 1070–1080. MIT Press, 1988.

[GL91] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.

[GNAJBD00] M.A. Gutiérrez Naranjo, J.A. Alonso Jiménez, and J. Borrego Díaz. A topological study of the upward refinement operators in ILP. In J. Cussens and A. Frisch, editors, *Inductive Logic Programming, 10th International Conference (ILP2000) Work-in-Progress Reports, London, UK*, pages 120–137, 2000.

[GRS91] A. Van Gelder, K.A. Ross, and J.S. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):620–650, 1991.

[Hec99] R. Heckmann. Approximation of metric spaces by partial metric spaces. *Applied Categorical Structures*, 7:71–83, 1999.

[HK94] S. Hölldobler and Y. Kalinke. Towards a massively parallel computational model for logic programming. In *Proc. ECAI94 Workshop on Combining Symbolic and Connectionist Processing*, pages 68–77. ECCAI, 1994.

[HS99a] P. Hitzler and A.K. Seda. Characterizations of classes of programs by three-valued operators. In M. Gelfond, N. Leone, and G. Pfeifer, editors, *Logic Programming and Nonmonotonic Reasoning, Proceedings of the 5th International Conference on Logic Programming and Non-Monotonic Reasoning (LPNMR'99), El Paso, Texas, USA*, volume 1730 of *Lecture Notes in Artificial Intelligence*, pages 357–371. Springer, Berlin, 1999.

[HS99b] P. Hitzler and A.K. Seda. The fixed-point theorems of Priess-Crampe and Ribenboim in logic programming. In *Valuation Theory and its Applications, Proceedings of the 1999 Valuation Theory Conference, University of Saskatchewan in Saskatoon, Canada*, Fields Institute Communications Series. American Mathematical Society, 1999. 17 pages, to appear.

[HS99c] P. Hitzler and A.K. Seda. Some issues concerning fixed points in computational logic: Quasi-metrics, multivalued mappings and the Knaster-Tarski theorem. In *Proceedings of the 14th Summer Conference on Topology and its Applications: Special Session on Topology in Computer Science, New York*, volume 24 of *Topology Proceedings*, 1999. 18 pages, to appear.

[HS00] P. Hitzler and A.K. Seda. A new fixed-point theorem for logic programming semantics. In *Proceedings of the joint IIIS & IEEE meeting of the 4th World Multiconference on Systemics, Cybernetics and Informatics (SCI2000) and the 6th International Conference on Information Systems Analysis and Synthesis (ISAS2000), Orlando, Florida, USA*, volume VII, Computer Science and Engineering Part 1, pages 418–423. International Institute of Informatics and Systemics: IIIS, 2000.

[HSK99] S. Hölldobler, H. Störr, and Y. Kalinke. Approximating the semantics of logic programs by recurrent neural networks. *Applied Intelligence*, 11:45–58, 1999.

[HSW89] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[KKM93] M.A. Khamsi, V. Kreinovich, and D. Misane. A new method of proving the existence of answer sets for disjunctive logic programs: A metric fixed-point theorem for multivalued mappings. In C. Baral and M. Gelfond, editors, *Proceedings of the Workshop on Logic Programming with Incomplete Information, Vancouver, B.C., Canada*, pages 58–73, 1993.

[KM98] M.A. Khamsi and D. Misane. Disjunctive signed logic programs. *Fundamenta Informaticae*, 32(3–4):349–357, 1997/1998.

[Kuh99] F.V. Kuhlmann. A theorem about maps on spherically complete ultrametric spaces, and its applications. Preprint, Department of Mathematics and Statistics, University of Saskatchewan in Saskatoon, 20 pages, 1999.

[Kun87] K. Kunen. Negation in logic programming. *Journal of Logic Programming*, 4:289–308, 1987.

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

[LNS82] A.-L. Lassez, V.L. Nguyen, and E.A. Sonenberg. Fixed point theorems and semantics: A folk tale. *Information Processing Letters*, 14(3):112–116, 1982.

[Mah88] M. Maher. Equivalences of logic programs. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*. Morgan Kaufmann, Los Altos, CA, 1988.

[Mar95] E. Marchiori. A methodology for proving termination of general logic programs. In *Proc. 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 356–367, 1995.

[Mar96] E. Marchiori. On termination of general logic programs with respect to constructive negation. *Journal of Logic Programming*, 26(1):69–89, 1996.

[Mat86] S.G. Matthews. Metric domains for completeness. Technical Report 76, Department of Computer Science, University of Warwick, UK, April 1986. Ph.D. Thesis, 1985.

[Mat92] S.G. Matthews. The cycle contraction mapping theorem. Technical Report 228, Department of Computer Science, University of Warwick, UK, 1992.

[Mat94] S.G. Matthews. Partial metric topology. In *Proceedings of the Eighth Summer Conference on General Topology and its Applications*, volume 728 of *Annals of the New York Academy of Sciences*, pages 183–197, 1994.

[Mat99] C. Mateis. Extending disjunctive logic programming by T-norms. In M. Gelfond, N. Leone, and G. Pfeifer, editors, *Logic Programming and Nonmonotonic Reasoning, Proceedings of the 5th International Conference on Logic Programming and Non-Monotonic Reasoning (LPNMR'99), El Paso, Texas, USA*, volume 1730 of *Lecture Notes in Artificial Intelligence*, pages 290–304. Springer, Berlin, 1999.

[MT99] V.M. Marek and M. Truszczyński. Stable models and an alternative logic programming paradigm. In K.R. Apt, V.W. Marek, M. Truszczyński, and D.S. Warren, editors, *The Logic Programming Paradigm: A 25 Year Persepective*, pages 375–398. Springer, Berlin, 1999.

[Myc84] A. Mycroft. Logic programs and many-valued logic. In M. Fontet and E. Mehlhorn, editors, *STACS 84, Symposium of Theoretical Aspects of Computer Science, Paris, France, 1984, Proceedings*, volume 166 of *Lecture Notes in Computer Science*, pages 274–286. Springer, 1984.

[Nai98] L Naish. A three-valued semantics for Horn clause programs. Technical Report 98/4, University of Melbourne, 1998.

[O'N95] J. O'Neill. Two topologies are better than one. Technical Report CS-RR-283, Department of Computer Science, University of Warwick, April 1995.

[PC90] S. Prieß-Crampe. Der Banachsche Fixpunktsatz fur ultrametrische Räume. *Results in Mathematics*, 18:178–186, 1990.

[PCR93] S. Prieß-Crampe and P. Ribenboim. Fixed points, combs and generalized power series. *Abh. Math. Sem. Univ. Hamburg*, 63:227–244, 1993.

[PCR00a] S. Prieß-Crampe and P. Ribenboim. Fixed point and attractor theorems for ultrametric spaces. *Forum Math.*, 12:53–64, 2000.

[PCR00b] S. Prieß-Crampe and P. Ribenboim. Logic programming and ultrametric spaces. *Rendiconti di Mathematica*, VII:1–13, 2000. To appear.

[PCR00c] S. Prieß-Crampe and P. Ribenboim. Ultrametric spaces and logic programming. *Journal of Logic Programming*, 42:59–70, 2000.

[PP90] H. Przymusinska and T.C. Przymusinski. Weakly stratified logic programs. *Fundamenta Informaticae*, 13:51–65, 1990. K.R. Apt, editor, special issue of Fundamenta Informaticae on Logical Foundations of Artificial Intelligence.

[Prz88] T.C. Przymusinski. On the declarative semantics of deductive databases and logic programs. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 193–216. Morgan Kaufmann, Los Altos, CA, 1988.

[PZ98] F. Protti and G. Zaverucha. On the relations between acceptable programs and stratifiable classes. In F.M. De Oliveira, editor, *Advances in Artificial Intelligence, 14th Brazilian Symposium on Artificial Intelligence, SBIA '98, Porto Alegre, Brazil*, volume 1515 of *Lecture Notes in Computer Science*, pages 141–150. Springer, Berlin, 1998.

[Rib96] P. Ribenboim. The new theory of ultrametric spaces. *Periodica Mathematica Hungarica*, 32(1–2):103–111, 1996.

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

[RZ98] W.C. Rounds and G.-Q. Zhang. Clausal logic and logic programming in algebraic domains. Preprint, submitted to Information and Computation, 21 pages, 1998.

[Sed95] A.K. Seda. Topology and the semantics of logic programs. *Fundamenta Informaticae*, 24(4):359–386, 1995.

[Sed97] A.K. Seda. Quasi-metrics and the semantics of logic programs. *Fundamenta Informaticae*, 29(1):97–117, 1997.

[SH97] A.K. Seda and P. Hitzler. Topology and iterates in computational logic. In *Proceedings of the 12th Summer Conference on Topology and its Applications: Special Session on Topology in Computer Science, Ontario*, volume 22 of *Topology Proceedings*, pages 427–469, August 1997.

[SH99] A.K. Seda and P. Hitzler. Strictly level-decreasing logic programs. In A. Butterfield and S. Flynn, editors, *Proceedings of the Second Irish Workshop on Formal Methods (IWFM'98), Cork, Ireland*, Electronic Workshops in Computing (eWiC), pages 1–18. British Computer Society, 1999.

[SHLG94] V. Stoltenberg-Hansen, I. Lindstrom, and R. Griffor. *Mathematical Theory of Domains*. Cambridge University Press, 1994.

[Smy87] M.B. Smyth. Quasi uniformities: Reconciling domains with metric spaces. In M. Main, A. Melton, M. Mislove, and D. Schmidt, editors, *Mathematical Foundations of Programming Language Semantics*, volume 198 of *Lecture Notes in Computer Science*, pages 236–253. Springer, Berlin, 1987.

[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, editors, *Topology and Category Theory in Computer Science*, pages 207–229. Oxford University Press, 1991.

[ŠŠ82] J. Šebelík and P. Štěpánek. Horn clause programs for recursive functions. In K.L. Clark and S.-Å. Tarnlund, editors, *Logic Programming*, pages 324–340. Academic Press, New York, 1982.

[Wac00] P. Waczkiewicz. Measurements and weak partial metrics. Technical Report CSR-00-11, School of Computer Science, The University of Birmingham, 2000.

[Wil70] S. Willard. *General Topology*. Addison-Wesley, Reading, MA, 1970.

[Zha99] Y. Zhang. On logical semantics of hybrid symbolic-neural networks for commonsense reasoning. In *Proceedings of the 1999 IEEE International Conference on Neural Networks*, pages 502–505. IEEE, 1999.

[ZR97a] G.-Q. Zhang and W.C. Rounds. Complexity of power default reasoning. In *Proceedings of the Twelfth Annual IEEE Symposium on Logic in Computer Science, LICS'97, Warsaw, Poland*, pages 328–339. IEEE Computer Society Press, 1997.

[ZR97b] G.-Q. Zhang and W.C. Rounds. Power defaults (preliminary report). In J. Dix, U. Furbach, and A. Nerode, editors, *Proceedings* *of the Fourth International Conference on Logic Programming and Non-Monotonic Reasoning (LPNMR'97), Dagstuhl, Germany*, volume 1265 of *Lecture Notes in Computer Science*, pages 152–169. Springer, 1997.

[ZR98] G.-Q. Zhang and W.C. Rounds. Semantics of logic programs and representation of Smyth powerdomains. Preprint, 1998.

Index of Definitions

I hereby declare that this is my own work and that it has not been submitted for the award of any degree at any other university.

Date:

Signature: