# Multivalued Mappings, Fixed-Point Theorems and Disjunctive Databases

Pascal Hitzler*
Department of Mathematics, National University of Ireland - Cork,
Cork, Ireland
Email: phitzler@gmx.de

Anthony Karel Seda
Department of Mathematics, National University of Ireland - Cork,
Cork, Ireland
Email: aks@ucc.ie

**Abstract**

In this paper, we discuss the semantics of disjunctive programs and databases and show how multivalued mappings and their fixed points arise naturally within this context. A number of fixed-point theorems for multivalued mappings are considered, some of which are already known and some of which are new. The notion of a normal derivative of a disjunctive program is introduced. Normal derivatives are normal logic programs which are determined by the disjunctive program. Thus, the well-known single-step operator associated with a normal derivative is single-valued, and its fixed points can be found by well-established means. It is shown how fixed points of the multivalued mapping determined by a disjunctive program relate to the fixed points of the single-step operators coming from its normal derivatives. This procedure has potential for simplifying the construction of models of disjunctive databases, and this point is discussed.

Most of the results for multivalued mappings rest on corresponding, known results concerning fixed points of single-valued mappings. Since the latter results are frequently referred to, they have been collected together for convenience in a survey which should be of independent interest as well as being preparatory for the later results. Finally, a number of problems and issues raised by this work are discussed.

## 1 Introduction

Let $f : X \to X$ be a (single-valued) function or mapping defined on the set $X$. A *fixed point* of $f$ is an element $x$ of $X$ such that $f(x) = x$. This simple, but important, notion is to be found in many places in mathematics ranging from proofs of existence of solutions of differential equations to proofs of existence of invariant measures to methods, in logic, for handling self-reference. Indeed, it is this latter use of "fixed point" that makes the concept so fundamental in programming language semantics and in program correctness: the meaning of recursive definitions of functions and of inductive definitions of sets is the least fixed point of an operator naturally associated with the definition. This last comment applies equally well, of course, whether one is taking the point of view of imperative programming or the point of view of declarative programming. In theoretical studies in either paradigm, fixed points of functions and operators are basic.

*The first named author acknowledges financial support under grant SC/98/621 from Enterprise Ireland.

It follows naturally from what has just been said that techniques for finding fixed points, i.e. fixed-point theorems, are also basic in theoretical studies. In the imperative case, the functions and operators which arise are nearly always monotonic, and therefore the only fixed-point theorem one needs is “the fixed-point theorem” of Knaster and Tarski. However, in logic programming matters are not so simple due to the presence of negation. Negation has the effect of introducing non-monotonicity and of complicating the process of finding fixed points. Thus, within this paradigm, one finds quite a variety of different methods and techniques in use for finding fixed points of the operators arising from questions concerned with semantics.

The purpose of this paper is three-fold. First, in Section 3, we want to report on recent work undertaken by the authors in the context of the semantics of disjunctive programs and databases. In this setting, the operators $T$ which arise are *multivalued* i.e. $T : X \to 2^X$ so that $T(x)$ is a set of points of $X$, rather than a single point. The usual meaning of “fixed point” is then an element $x \in X$ such that $x \in T(x)$, and this clearly generalizes the case of single-valued mappings. However, not only is this mathematically an obvious generalization of the case of single-valued mappings, it is also a correct one in that issues concerned with the semantics of disjunctive programs naturally relate to fixed points in this new sense. Thus, there is a need for fixed-point theorems which apply to multivalued mappings. However, one strategy which we introduce here is to extract from the database a family of associated conventional normal logic programs (which we call normal derivatives) whose corresponding operators are single-valued. In this way, it is possible to relate the fixed points of multivalued mappings to the fixed points of single-valued mappings. We demonstrate the success of this method by showing how the stable model semantics (or answer set semantics) of Gelfond and Lifschitz can be treated in this way. In fact, there is a whole range of semantics which have been proposed for disjunctive databases, of which that of Gelfond and Lifschitz is just one of the more important, and all of which provide different, canonical models of a given database (perfect model, weakly perfect model etc.). Each of these models turns out to be a fixed point of a certain multivalued mapping we introduce here (the single-step operator of Definition 3.3) and hence, as we show, is a fixed point of some normal derivative. This, of course, raises the problem of characterizing those normal derivatives which correspond to any particular canonical model, and is a problem under investigation by the authors.

The alternative approach to finding fixed points, already noted, is the full-frontal attack by means of fixed-point theorems for multivalued mappings. However, there does not appear to be very many such theorems available, and it is ongoing work of the authors to try and rectify this situation either by extending results known for single-valued mappings to the multivalued case or by establishing new methods and techniques. Thus, in Section 4, we collect together, as our second objective, the main results we know and which we know to have applications to semantics. In addition, we present some new technical results of our own, and discuss the problems and difficulties in the way to further progress.

Needless to say, much of the work undertaken in Section 4 rests on well-established results and ideas developed specifically for the single-valued case. It will facilitate the discussion to have available the statements of the more important of these, and we collect them together in a survey in Section 2 for convenience and for reference, relating them as we proceed to applications of our own, and of others, to semantics; this is our third main objective in the paper. Therefore, in stucture, we see Section 3 as the heart of the paper, but that the material presented there immediately raises the issues taken up in Section 4 and, in turn, this latter section depends on the discussion of Section 2. The order of presentation of the material is chosen, of course, to enable us to proceed from the known to the unknown.

# 2 Fixed-Point Theorems for Single-Valued Mappings

## 2.1 Partial Orders

Perhaps the original theorem in the subject is Kleene's first recursion theorem, encountered in recursive function theory. This states the following, where $\mathcal{F}_n$ denotes the collection of all partial functions from $\mathbb{N}^n$ to $\mathbb{N}$ ordered by graph inclusion. Suppose that $\Phi : \mathcal{F}_n \to \mathcal{F}_n$ is a recursive operator. Then $\Phi$ has a least fixed point $h$ which is a computable function. Thus, there is a computable function $h$ satisfying the following conditions.
(a) $\Phi(h) = h$.
(b) If $\Phi(g) = g$, then $h \leq g$.
Hence, if $h$ is total, it is the only fixed point of $\Phi$.

The term *recursive* used here means, essentially, that whenever $\Phi(f)(\mathbf{x})$ is defined, its value depends only on finitely many values of $f$. Of course, as stated, this theorem is of limited applicability but has a direct generalization to partially ordered sets and continuous functions, as follows.

**2.1 Definition** A subset $S$ of a partially ordered set $(D, \leq)$ is called *directed* if, for any $a, b \in S$, there is $c \in S$ such that $a \leq c$ and $b \leq c$.

**2.2 Definition** A partially ordered set $(D, \leq)$ is said to be *complete*, and hence a *complete partial order* (cpo), if there is a least element of $D$ and every directed subset $S \subseteq D$ has a least upper bound $\sup S$ in $D$. Thus:

(1) There is an element $\perp$ of $D$ (the bottom element of $D$) such that $\perp \leq d$ for all $d \in D$.

(2) If $S \subseteq D$ is directed, then there is an element $\sup S$ in $D$ such that (i) $s \leq \sup S$ for all $s \in S$, and (ii) if $s \leq d$ for all $s \in S$, where $d$ is some element of $D$, then $\sup S \leq d$.

**2.3 Definition** Let $D$ and $E$ be 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 *continuous* if $f$ is monotonic and, for every directed subset $S$ of $D$, we have $f(\sup S) = \sup f(S)$.

We are now in a position to state the main fixed-point theorem applicable to partially ordered sets, see [22] for a proof; it is the promised generalization of Kleene's theorem[1]. In fact, it can be established in the context of the slightly more general $\omega$-cpos, which are defined as above except that one only requires the existence of the suprema of increasing sequences $a_0 \leq a_1 \leq a_2 \ldots$ in $D$, rather than of directed sets.

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

This theorem has so many applications to computing that it must be a contender for the title of fundamental theorem of computer science. In addition, there are numerous refinements and variants of it scattered throughout the literature. One such is the following theorem, which was applied by Fitting in the context of logic programming semantics over Kleene's 3-valued logic.

[1] In so far as recursiveness of $\Phi$, and the existence of its least fixed point are concerned. The question of generalizing the computability of the fixed point provided by Kleene's theorem depends on abstract versions of computability theory, and is a topic considered in the literature in domain theory.

**2.5 Definition** Let $(D, \leq)$ be a partial order. Then $D$ is called a *complete semilattice* if the following conditions hold.

1. Every non-empty subset of $D$ has an infimum.
2. Every non-empty directed subset of $D$ has a supremum.

**2.6 Theorem** (See [3, Proposition 6.2]) Let $(D, \leq)$ be a complete semilattice, let $\Phi$ be monotonic on $D$ and let $a \in D$ be such that $a \leq \Phi(a)$. Then $\Phi$ has a least fixed point above $a$.

The least fixed point in the previous theorem can, in fact, be obtained as an ordinal power $\Phi \uparrow \alpha(a)$ for some ordinal $\alpha$.

Although we know of no actual application of the next result, see [2, Theorem 1.4.1], it is possible that it has some role, since greatest fixed points are important in computational logic in connection with implementation issues such as negation as failure. In any case, we give a generalization of it to multivalued mappings in Section 4, where we consider generalizations of the Knaster-Tarski theorem as well.

**2.7 Theorem** Let $(D, \leq)$ be a partially ordered set such that every chain in $D$ has a supremum, let $f : D \to D$ be monotonic and let $a \in D$ be such that $a \leq f(a)$. Then $f$ has a maximal fixed point.

## 2.2 Metric-Like Spaces

The other great fixed-point theorem in mathematics is the well-known Banach contraction mapping theorem. Apart from its applications to the semantics of concurrency and communicating systems, it does not seem to be much used in studying imperative languages. By contrast, it and its generalizations have found quite a lot of interest in logic programming, due to the non-monotonicity introduced by negation, and we consider these next.

### *2.2.1 Metric Spaces*

**2.8 Theorem (Banach Contraction Mapping Theorem)** Let $(X, d)$ be a complete metric space, let $0 < \lambda < 1$ and let $f : X \to X$ be a function satisfying $d(f(x), f(y)) \leq \lambda d(x, y)$ for all $x, y \in X$. Then $f$ has a unique fixed point which can be obtained as the limit of the sequence $f^n(x)$ for any $x \in X$.

Note that the proof is constructive, i.e. the fixed point is in fact the limit of any sequence of iterates of $f$.

The Banach theorem has found application to logic programming in [4, 19, 20], and a multivalued version was considered in [10] and will be discussed in Section 4. In fact, quite a lot of work has been done, some of it by the present authors, in applying generalizations of the Banach theorem in which the axioms in the definition of a metric are relaxed, see [14, 19, 20], and we briefly consider this next.

### *2.2.2 Generalized Ultrametric Spaces*

The first generalization we consider, and it is a significant one, is obtained by allowing a metric to take values in an arbitrary partially ordered set, rather than just in the real numbers.

**2.9 Definition** Let $X$ be a set and let $\Gamma$ be a partially ordered set with least element $0$. We call $(X, d)$ a *generalized ultrametric space* if $d : X \times X \to \Gamma$ is a function such that for all $x, y, z \in X$ and $\gamma \in \Gamma$

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

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

(3) if $d(x, y), d(y, z) \leq \gamma$, then $d(x, z) \leq \gamma$.

For $0 \neq \gamma \in \Gamma$ and $x \in X$, the set $B_\gamma(x) = \{y \in X \mid d(x, y) \leq \gamma\}$ is called a ($\gamma$-)ball in $X$. A generalized ultrametric space is called *spherically complete* if, for any chain $(\mathcal{C}, \subseteq)$ of balls in $X$, we have $\bigcap \mathcal{C} \neq \emptyset$.

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

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

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

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

For the following, see [14], also cf. [7, Theorem 4.4].

**2.10 Theorem (Prieß-Crampe and Ribenboim)** 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.

This result has been applied to logic programming semantics in [14, 19, 20], and we next sketch the application we made of it in [19, 20].

For a countable ordinal $\gamma$, let $\Gamma_\gamma$ be the set $\{2^{-\alpha} \mid \alpha < \gamma\}$ of symbols $2^{-\alpha}$ with ordering $2^{-\alpha} < 2^{-\beta}$ if and only if $\beta < \alpha$.

**2.11 Definition** Let $D$ be a domain (i.e. a Scott domain with set $D_c$ of compact elements, see [22]), let $r : D_c \to \gamma$ be a function, called a *rank function*, and denote $2^{-\gamma}$ by 0. Define $d_r : D \times D \to \Gamma_{\gamma+1}$ by

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

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

It is straightforward to see that $(D, d_r)$ is indeed a generalized ultrametric space. Indeed, $(D, d_r)$ is spherically complete under the additional condition that for each $x \in D$ and for each ordinal $\alpha < \gamma$, the set $\{c \in \text{approx}(x) \mid r(c) < \alpha\}$ is directed whenever it is non-empty, where $\text{approx}(x)$ denotes the set of compact elements $c$ such that $c \leq x$.

**2.12 Theorem** (See [19]) Under the condition just stated, $(D, d_r)$ is spherically complete.

Using this theorem in conjunction with Theorem 2.10, we showed that a subclass of the locally stratified programs is both computationally adequate (i.e. can compute all partial recursive functions) and has a unique supported model. This subclass we called the strictly level-decreasing programs, and it is rather rare for a class of programs to satisfy both the properties just mentioned simultaneously.

[2] In [14], these functions were called *contractive*.

### *2.2.3 Quasi-Metric Spaces*

The next thing one can do to generalize the Banach theorem is to vary the axioms describing the notion of "metric", as already mentioned. There are numerous ways of doing this, but perhaps the most successful is the notion of quasi-metric when taken in conjunction with the associated theorem of Rutten and Smyth, see [16, 21].

**2.13 Definition** A set $X$ together with a function $d : X \times X \to \mathbb{R} \cup \{+\infty\}$ is called a *quasi-pseudo-metric space* if for all $x, y, z \in X$

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

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

If, furthermore, $d(x, y) = d(y, x) = 0$ implies $x = y$, then $(X, d)$ is called a *quasi-metric space*.

A quasi-pseudo-metric space in which the strong triangle inequality $d(x, y) \leq \max\{d(x, z), d(z, y)\}$ holds for all $x, y, z \in X$, is called a quasi-pseudo-*ultra*metric space. Consequently, a quasi-pseudo-ultrametric space which is a quasi-metric space is called a *quasi-ultrametric space*.

A sequence $(x_n)$ in $X$ is a (*forward-*) *Cauchy-sequence* (*CS*) if, for all $\varepsilon > 0$, there exists $n_0 \in \mathbb{N}$ such that for all $n \geq m \geq n_0$, $d(x_m, x_n) < \varepsilon$. A CS $(x_n)$ *converges to* $x \in X$ (written $x_n \to x$, or $\lim x_n = x$) if, for all $y \in X$, $d(x, y) = \lim d(x_n, y)$. Finally, $X$ is called *CS-complete* if every CS in $X$ converges.

Note that limits of CSs need not be unique. If $X$ is a quasi-metric space, however, it is a standard fact that uniqueness of limits does hold.

**2.14 Definition** Let $X$ be a quasi-pseudo-metric space. A function $f : X \to X$ is called

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

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

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

We are now in a position to state the main fixed-point theorem in the context of quasi-metric spaces. In this statement, $\leq_d$ is the partial order induced on $X$ by the quasi-metric $d$, where $x \leq_d y$ if and only if $d(x, y) = 0$.

**2.15 Theorem (Rutten-Smyth)** Let $X \neq \emptyset$ be a CS-complete quasi-metric 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 CS $(f^n(x))$, where in (1) $x$ is the given point, and in (2) $x$ can be chosen arbitrarily.

This theorem is important for several reasons, but we will take up just one of them. Let $(X, \leq)$ be a partial order. Define a function $d_{\leq} : X \times X \to \mathbb{R}$ by

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

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

By virtue of these definitions, Part (1) of Theorem 2.15 generalizes the Knaster-Tarski Theorem 2.4. Part (2) generalizes the Banach Contraction Mapping Theorem 2.8. Using this and related results, it was shown in [16, 21] that one can combine the two main approaches to the semantics of imperative programs i.e. the partial order approach and the metric approach. The same thing was done in [18] for logic programs, again using Theorem 2.15.

# 3 Multivalued Mappings and Disjunctive Programs

We now turn our attention to multivalued mappings. We will see how these naturally arise in questions of semantics in logic programming and database theory, by describing the answer set semantics for disjunctive programs and databases due to Gelfond and Lifschitz [6], see also [11] which we follow closely in giving the definition. In fact, our main results here, and indeed the main new results of this paper, relate the construction of the answer set semantics for a restricted class of disjunctive programs to the supported models of certain normal logic programs which are naturally associated with the disjunctive program. These associated programs we call "normal derivatives", and their supported models can be found by well-established means. Therefore, the process we put forward results in a considerable simplification of the construction of the answer set semantics for the class of disjunctive programs we consider.

## 3.1 The Answer Set Semantics of Gelfond and Lifschitz

Let $\mathtt{Lit}$ denote the set of ground literals in some first order language $\mathcal{L}$. In its most general form, a *rule* $r$ is an expression of the following type

$$l_1 \vee \cdots \vee l_n \leftarrow l_{n+1} \wedge \ldots, l_m \wedge \mathtt{not} l_{m+1} \wedge \cdots \wedge \mathtt{not} l_k,$$

where each $l_i \in \mathtt{Lit}$. Given such a rule $r$, we define $\mathtt{Head}(r) = \{l_1, \ldots, l_n\}$, $\mathtt{Pos}(r) = \{l_{n+1}, \ldots, l_m\}$ and $\mathtt{Neg}(r) = \{l_{m+1}, \ldots, l_k\}$. The keyword $\mathtt{not}$ may be interpreted either as negation as failure (in which case $\mathtt{Pos}(r)$ may contain negative literals) or as classical negation (in which case $\mathtt{Pos}(r)$ will contain only positive literals), although later on we restrict it to its latter meaning of classical negation. A rule $r$ is called *disjunctive* if $n \geq 1$, so that $\mathtt{Head}(r)$ may contain more than one element, and *non-disjunctive* otherwise. A (disjunctive) program $\Pi$ is a set of (disjunctive) rules. Of course, a non-disjunctive program is simply a conventional ground normal logic program. It is usual to allow the presence of function symbols in disjunctive programs, and to reserve the term disjunctive database for those disjunctive programs which do not contain function symbols, and we will observe this convention. Also, one may allow variable symbols to be present in the general definition of a rule. However, as implied by the definition we have adopted, we are in effect going to work with the set of ground instances of each rule, rather than with the rules themselves. Thus, the only difference between a disjunctive program and a disjunctive database is that in the former case we will be handling an infinite set of rules and in the latter case a finite set, and the distinction will not matter.

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

1. If $X$ contains complementary literals, then $X = \texttt{Lit}$.
2. $X$ is minimal i.e. if $A \subseteq X$ and $A$ is closed by rules of $\Pi$, then $A = X$.

We denote the set of answer sets of $\Pi$ by $\alpha(\Pi)$. If $\Pi$ is not disjunctive, then $\alpha(\Pi)$ is a singleton set. However, if $\Pi$ is disjunctive, then $\alpha(\Pi)$ may contain more than one element, and we give an example below to illustrate this phenomenon.

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

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

It is clear that the program $\Pi^X$ does not contain not and therefore $\alpha(\Pi^X)$ is defined. Following Gelfond and Lifschitz [6], we define the operator $GL : 2^{\texttt{Lit}} \to 2^{2^{\texttt{Lit}}}$ by $GL(X) = \alpha(\Pi^X)$. Finally, we say that $X$ is an *answer set* of $\Pi$ if $X \in \alpha(\Pi^X)$ i.e. if $X \in GL(X)$. In other words, $X$ is an answer set of $\Pi$ if it is a *fixed point* of the multivalued mapping $GL$. Notice that if $\Pi$ is not disjunctive, then $\alpha(\Pi^X)$ is a singleton set, as already observed, and so $X$ is an answer set of $\Pi$ if and only if $X = GL(X)$. The more general requirement that $X \in GL(X)$ is the natural, and standard, extension of the notion of fixed point to the case of multivalued functions and operators. Again, we use the notation $\alpha(\Pi)$ for the set of answer sets of $\Pi$ in the general case.

It will help to consider an example which illustrates these ideas.

**3.1 Example** Take $\Pi$ as follows:

$$p(0) \vee q(0) \leftarrow$$
$$p(a) \vee q(0) \leftarrow q(0) \wedge \neg p(0).$$

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

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

and the answer sets of $\Pi^X$ are $\{p(0)\}$ and $\{q(0)\}$. Thus, $\alpha(\Pi^X) = \{\{p(0)\}, \{q(0)\}\}$. Since $X = \{q(0)\}$ is an allowable value of $X$, we see that $X \in \alpha(\Pi^X)$ and hence that $\{q(0)\}$ is an answer set for $\Pi$.

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

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

Again, the only answer sets of $\Pi^X$ are $\{p(0)\}$ and $\{q(0)\}$. Since $X = \{p(0)\}$ is an allowable value of $X$ at the moment, we see that $\{p(0)\}$ is an answer set for $\Pi$, and indeed is the only one other than $\{q(0)\}$. Thus, $\alpha(\Pi) = \{\{p(0)\}, \{q(0)\}\}$.

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

### 3.2 Normal Derivatives of Disjunctive Logic Programs

In order to proceed, it will be necessary to restrict attention to a subclass of the disjunctive programs in which we allow only positive ground literals in the rules. 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.

**3.2 Lemma** Suppose that the head of each clause in a disjunctive program $\Pi$ contains only positive literals. Then any answer set for $\Pi$ contains only positive literals.

**Proof:** Suppose that $X$ is a set of literals which is closed by rules of $\Pi$. Let $Y$ denote the set which results by removing from $X$ all the negative literals in $X$. Then $Y$ is closed by rules of $\Pi$. To see this, suppose that $r \in \Pi$ and that $\texttt{Pos}(r) \subseteq Y$ is true. Then $\texttt{Pos}(r) \subseteq X$ is also true, and so $\texttt{Head}(r) \cap Y = \texttt{Head}(r) \cap X \neq \emptyset$.

Therefore, by minimality, an answer set of $\Pi$ can only contain positive literals. ■

Notice that this lemma makes redundant the condition 1. concerning complementary literals in the first part of the definition of an answer set.

Thus, for the rest of Section 3, the most general form of rule $r$ that we shall consider is the following

$$A_1 \vee \cdots \vee A_n \leftarrow B_{n+1} \wedge \cdots \wedge B_m \wedge \neg B_{m+1} \wedge \cdots \wedge \neg B_k,$$

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 [15]. We will continue to use the notation $\Pi$ for a typical disjunctive program even with this restriction in place. Hence, $\Pi$ denotes a possibly infinite set of rules of the sort just described.

The Lemma 3.2 focusses attention on the sets of positive ground literals in the first order language $\mathcal{L}$ underlying $\Pi$ i.e. on the power set $I_\Pi$ of the Herbrand base $B_\Pi$ of $\Pi$. We intend to relate answer sets to supported models of normal logic programs associated with $\Pi$, and Lemma 3.2 will assist us in doing this. Therefore, typical elements of $I_\Pi$ 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 $\Pi$ in the form $H_r \leftarrow \texttt{body}_r$.

**3.3 Definition** Suppose that $\Pi$ is a disjunctive logic program. The *single-step operator $T_\Pi$ associated with* $\Pi$ is the multivalued mapping from $I_\Pi$ to the power set $\mathcal{P}(I_\Pi)$ of $I_\Pi$ defined by: $J \in T_\Pi(I)$ if and only if the following conditions are satisfied.

(i) For each rule $H_r \leftarrow \texttt{body}_r$ in $\Pi$ such that $I \models \texttt{body}_r$ (i.e. $\texttt{body}_r$ is true with respect to $I$), there exists an $A$ in $H_r$ such that $A \in J$.

(ii) For all $A \in J$, there exists a rule $H_r \leftarrow \texttt{body}_r$ in $\Pi$ such that $I \models \texttt{body}_r$ 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 $\Pi$ is a normal logic program $P$.

**3.4 Theorem** Suppose that $\Pi$ is a disjunctive logic program. Then we have $I \in T_\Pi(I)$, i.e. $I$ is a fixed point of $T_\Pi$, if and only if the following conditions are satisfied.

(a) $I$ is a model for $\Pi$, i.e. for every rule $H_r \leftarrow \texttt{body}_r$ in $\Pi$ such that $\texttt{body}_r$ 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 $H_r \leftarrow \texttt{body}_r$ in $\Pi$ such that $\texttt{body}_r$ 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 $I_\Pi$) which fulfills Condition (b) above a *supported* interpretation. Thus, $I \in T_\Pi(I)$ if and only if $I$ is a supported model for $\Pi$.

**Proof:** Suppose that $I \in T_\Pi(I)$ and let $H_r \leftarrow \texttt{body}_r$ be a rule in $\Pi$ such that $\texttt{body}_r$ 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 3.3. Condition (b) follows directly from (ii) of Definition 3.3.

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 3.3 are satisfied for $I = J$. Both however follow directly from Conditions (a) and (b), respectively. ■

We now come to one of our main definitions.

**3.5 Definition** Suppose that $\Pi$ is a disjunctive logic program. A *normal derivative* $P$ of $\Pi$ 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 $H_r \leftarrow \texttt{body}_r$ in $\Pi$ there exists a clause $A \leftarrow \texttt{body}_r$ in $P$ such that $A$ belongs to $H_r$.

(b) For every clause $A \leftarrow \texttt{body}_r$ in $P$ there is a rule $H_r \leftarrow \texttt{body}_r$ in $\Pi$ such that $A$ belongs $H_r$.

Note that Condition (b) simply states that all clauses in $P$ have to be derived from rules in $\Pi$ by Condition (a).

**3.6 Theorem** Let $\Pi$ be a disjunctive logic program and let $I \in I_\Pi$. Then $J \in T_\Pi(I)$ iff $J = T_P(I)$ for some normal derivative $P$ of $\Pi$.

**Proof:** Let $P$ be a normal derivative of $\Pi$ 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 3.3.

For (i), let $H_r \leftarrow \texttt{body}_r$ be a rule in $\Pi$ such that $\texttt{body}_r$ is true with respect to $I$. By Condition (a) of the previous definition, there exists a clause $A \leftarrow \texttt{body}_r$ 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 $A \leftarrow \texttt{body}$ in $P$ such that $\texttt{body}$ is true with respect to $I$. By Condition (b) of the previous definition, there exists a rule $H \leftarrow \texttt{body}$ in $\Pi$ 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 3.3. We have to show that there exists a normal derivative $P$ of $\Pi$ such that $J = T_P(I)$. To do this, we define the ground normal program $P$ as follows.

(1) Let $H_r \leftarrow \texttt{body}_r$ be a rule in $\Pi$ such that $\texttt{body}_r$ 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 $A \leftarrow \texttt{body}_r$ for such $A$.

(2) For every rule $H_r \leftarrow \texttt{body}_r$ in $\Pi$ such that $\texttt{body}_r$ is not true with respect to $I$, we choose an atom $A$ in $H_r$ arbitrarily. Let $P$ contain all clauses $A \leftarrow \texttt{body}_r$ thus defined.

(3) $P$ contains only clauses defined by (1) and (2).

Obviously, $P$ is a normal derivative of $\Pi$.

Now let $A \in J$. Then by (1) there exists a clause $A \leftarrow \texttt{body}$ in $P$ such that $\texttt{body}$ is true with respect to $I$. Consequently, $A \in T_P(I)$. Conversely, let $A \in T_P(I)$. Then there is a clause $A \leftarrow \texttt{body}$ in $P$ such that $\texttt{body}$ is true with respect to $I$. By (1) and (3) there exists a rule $H \leftarrow \texttt{body}$ in $\Pi$ such that $A$ belongs to $H$, and by (1) again, we obtain $A \in J$ as required. ■

**3.7 Important Remark** The previous theorem allows us to conclude the existence of supported models for any given disjunctive program $\Pi$ provided any normal derivative of $\Pi$ has such a model. In particular, if any normal derivative of $\Pi$ is acceptable, see [1, 4, 8], or strictly level-decreasing, see [20], or locally stratified, see [15, 19] or definite, then $\Pi$ has at least one supported model. Conversely, if a given disjunctive program $\Pi$ has a supported model, there exists a normal derivative of $\Pi$ which has a supported model. This fact is important from our point of view since we are focussing on normal derivatives of $\Pi$ in the belief that they simplify the study of $\Pi$.

Recall that a disjunctive database $\Pi$ is a disjunctive logic program $\Pi$ without any function symbols. Thus, $\Pi$ consists of only finitely many rules in this case, and we denote their number by $n_\Pi$ and call it the *order* of $\Pi$.

**3.8 Proposition** Let $\Pi$ be a disjunctive database of order $n_\Pi = n$ consisting of the rules $r_1, r_2, \dots r_n$. For every $k \in \{1, \dots, n\}$, let $d_k$ denote the number of disjunctions occurring in the head of $r_k$. Then $\Pi$ has at most $\prod_{k=1}^{n}(2^{d_k} - 1)$ normal derivatives. Therefore, for any $I \in I_\Pi$ we have $|T_\Pi(I)| \leq \prod_{k=1}^{n_\Pi}(2^{d_k} - 1)$.

**Proof:** Let $r_k$ be a rule in $\Pi$. Every normal derivative $P$ of $\Pi$ contains at least one and at most $d_k$ clauses generated by $r_k$. Consequently, there are $\sum_{m=1}^{d_k}\binom{m}{d_k} = \left(\sum_{m=0}^{d_k}\binom{m}{d_k}\right) - \binom{0}{d_k} = 2^{d_k} - 1$ 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 3.6. ■

**3.9 Remark** For any disjunctive database which happens to be a normal logic program, the bound in the previous proposition turns out to be 1, so that this bound is sharp.

## 3.3 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 $\Pi$ and supported models of normal derivatives of $\Pi$.

**3.10 Theorem** Suppose that $\Pi$ is a disjunctive logic program in which $\mathtt{Head}(r)$ contains only positive literals for each rule $r \in \Pi$, and in which `not` denotes classical negation. Then given an answer set $X \in 2^{\mathtt{Lit}}$ for $\Pi$, there is a normal derivative $P$ of $\Pi$ such that $T_P(X) = X$.

**Proof:** We have $X \in \alpha(\Pi^X)$. Consider $\Pi^X$ and the following normal derivative $P$ of $\Pi$ which we construct by reference to the step by step construction of $\Pi^X$. Let $r$ be rule in $\Pi$ and suppose for ease of notation that $r$ takes the form $H_r \leftarrow \mathtt{body}_r$.

First, suppose that $\mathtt{Neg}(r) \cap X \neq \emptyset$, so that $r \notin \Pi^X$. We choose an atom, $A$ say, from the head $H_r$ of $r$ arbitrarily and include the clause $A \leftarrow \mathtt{body}_r$ in $P$. Since $\mathtt{Neg}(r) \cap X \neq \emptyset$ we see that $X \not\models \mathtt{body}_r$, 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 $\Pi^X$, 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 $\Pi^X$, we have the statement $\mathtt{Pos}(r') \subseteq X \Rightarrow \mathtt{Head}(r') \cap X \neq \emptyset$ holding true. The first subcase of this case is when $\mathtt{Pos}(r') \not\subseteq X$. 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 $\mathtt{Pos}(r) = \mathtt{Pos}(r') \not\subseteq X$, we have $X \not\models \mathtt{body}_r$ once more. Therefore, this clause also contributes nothing to $T_P(X)$.

Finally, consider the subcase of the previous case in which $\mathtt{Pos}(r') \subseteq X$, so that $\mathtt{Pos}(r) = \mathtt{Pos}(r') \subseteq X$. 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 $\mathtt{Pos}(r) \subseteq X$ 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 $\mathtt{Pos}(r) \subseteq X$. Therefore, we have $T_P(X) \subseteq X$, and $P$ is a normal derivative of $\Pi$ 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 $\Pi$ with $\mathtt{Pos}(r) \subseteq X$ we have $x \notin X \cap \mathtt{Head}(r)$. We show that this supposition leads to the contradiction that $Y = X \setminus \{x\} \subset X$ is an answer set for $\Pi^X$. Indeed, if $r$ is a rule in $\Pi^X$ such that $\mathtt{Pos}(r) \subseteq Y$, then $\mathtt{Pos}(r) \subseteq X$ and so $\mathtt{Head}(r) \cap Y = \mathtt{Head}(r) \cap X \neq \emptyset$. Thus, $Y$ is closed by rules of $\Pi^X$. But this contradicts the minimality of $X$ and concludes the proof. ■

As an immediate corollary of our results, we can recover the result of Gelfond and Lifshitz that an answer set for $\Pi$ is a model for $\Pi$ (and hence the name answer set semantics or *stable model semantics*).

**3.11 Corollary** Suppose that $\Pi$ is a disjunctive logic program. Then any answer set $X$ for $\Pi$ is a model for $\Pi$.

**Proof:** By the previous theorem, there is a normal derivative $P$ of $\Pi$ such that $T_P(X) = X$. Therefore, we have $X \in T_\Pi(X)$ by Theorem 3.6. It now follows that $X$ is a supported model for $\Pi$ by Theorem 3.4. ■

The normal derivative $P$ constructed in the proof just given of Theorem 3.10 has the maximality property that, for each rule $r \in \Pi$ which satisfies $\mathtt{Pos}(r) \subseteq X$, $P$ contains the clause $A \leftarrow \mathtt{body}_r$ for every atom $A \in \mathtt{Head}(r) \cap X$. One might say that $P$ is *maximal with respect to* $X$ whenever it satisfies this property, and it means that the normal derivative $P$ just constructed is in a sense canonical. Of course, the question of establishing a converse result arises, and as a first step in this direction we prove the following proposition.

**3.12 Proposition** Suppose that $\Pi$ is a disjunctive logic program which satisfies the hypothesis of the previous theorem. Suppose also that $X \in 2^{\mathtt{Lit}}$ and that $P$ is a normal derivative of $\Pi$ such that $T_P(X) = X$. Then $X$ is closed by rules of $\Pi^X$.

**Proof:** Let $r' \in \Pi^X$ be an arbitrary rule. Then there is a rule $r$ in $\Pi$ 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 $\mathtt{Pos}(r') \subseteq X$. Then $\mathtt{Pos}(r) \subseteq X$ and therefore $X \models \mathtt{body}_r$, since $\mathtt{Neg}(r) \cap X = \emptyset$. But $P$ is a normal derivative of $\Pi$ 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, $\mathtt{Head}(r') \cap X = \mathtt{Head}(r) \cap X \neq \emptyset$. Thus, $X$ is closed by rules of $\Pi^X$ as stated. ■

Thus, in the circumstances of the proposition just established, $X$ will be an answer set for $\Pi$ if and only if it is minimal with the property that it is a supported model of some normal derivative of $\Pi$. As already noted in the Introduction, this raises the problem of characterizing those normal derivatives whose fixed points are answer sets for $\Pi$. Indeed, the same problem can be put for all the other semantics which have been proposed for disjunctive programs and databases, and these questions will be pursued elsewhere.

# 4 Fixed-Point Theorems for Multivalued Mappings

Despite the possibilities presented by considering normal derivatives, the more obvious approach to finding fixed points of multivalued mappings is to employ fixed-point theorems applicable to them. Thus, in this section we discuss precisely this topic, building on the results we surveyed in Section 2. However, the treatment will necessarily be less complete than that in Section 2 because far fewer results appear to be known for multivalued mappings than for single-valued mappings, and thus we can ask more questions than we can answer. We shall follow the same order of presentation as in Section 2, beginning with theorems that depend on partial orders, and moving progressively through the use of metrics, generalized ultrametrics and, finally, quasi-metric spaces.

## 4.1 Partial Orders

The first result we consider is as follows. It is a generalization of the Knaster-Tarski theorem, Theorem 2.4, and was proved in [11].

**4.1 Definition** Let $(L, \leq)$ be a complete lattice. Define the preorder $\prec_r$ in $2^L$ by

$$A \prec_r B \text{ iff for all } y \in B \text{ there exists } x \in A \text{ such that } x \leq y.$$

A multivalued mapping $T : L \to 2^L$ is $\prec_r$*-increasing* if $x \leq y$ implies $T(x) \prec_r T(y)$ for all $x, y$.

Let $L$ be an ordered set and let $T : L \to 2^L$. We say that the family $(x_\beta)$ is a *decreasing $T$-orbit* if $x_{\beta+1} \in T(x_\beta)$ and $x_{\beta+1} \leq x_\beta$.

**4.2 Theorem** Let $L$ be a complete lattice and let $T$ be a $\prec_r$-increasing multivalued mapping from $L$ into $2^L$ satisfying the following two conditions.

1. For every $x \in L$, the set $T(x)$ is non-empty.
2. For every decreasing $T$-orbit $(x_\beta)$, there exist $x \in L$ such that $x \in T(\inf x_\beta)$ and $x \leq x_\beta$ for all $\beta$.

Then $T$ has a fixed point, i.e. there exists $x \in L$ such that $x \in T(x)$.

This theorem was established in [11] in order to show the existence of (consistent) answer sets for a class of disjunctive programs called signed programs. This class includes examples related to the well-known Yale Shooting Domain.

The following result is new, as far as we know, and is a generalization of Theorem 2.7.

**4.3 Theorem** Let $(D, \leq)$ be a partial order such that every chain in $D$ has a supremum and let $f : D \to 2^D$ be a multivalued mapping which is *monotonic* in the sense that whenever $x \leq y$ and $a \in f(x)$ then there exists some $b \in f(y)$ with $a \leq b$. Furthermore, let there be some $x_0 \in D$ such that there exists some $x_1 \in f(x_0)$ with $x_0 \leq x_1$. Then $f$ has a fixed point.

**Proof:** If $x_0 = x_1$ then it is a fixed point. So assume $x_0 < x_1$. Define an increasing chain $(x_\beta)$ in $D$ as follows. Let $x_\beta$ be defined for all $\beta < \alpha$. If $\alpha = \beta + 1$ is a successor ordinal, then choose $x_\alpha \in f(x_\beta)$ provided $x_\beta$ is not a fixed point of $f$ (since otherwise a fixed point is already found). If $\alpha$ is a limit ordinal, then let $x_\alpha = \sup_{\beta<\alpha}\{x_\beta\}$.

The increasing chain $(x_\alpha)$ has a supremum $s$, and by construction of $(x_\alpha)$ we must have $s \in f(s)$. ∎

## 4.2 Metric-Like Spaces

Just as in Section 2, the main alternative to order-theoretic arguments in the present context is to use various metric-like structures, and we consider these next.

### *4.2.1 Metric Spaces*

The following definition is standard.

**4.4 Definition** Let $(M, d)$ be a metric space. A multivalued mapping $T : M \to 2^M$ is called a *contraction* if there exists a real number $k < 1$ such that for every $x \in M$, for every $y \in M$, 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 [10], and depends on an old result of S.B. Nadler.

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

Again, 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.

### *4.2.2 Generalized Ultrametric Spaces*

**4.6 Definition** Let $(X, d, \Gamma)$ be a generalized ultrametric space (so that $\Gamma$ is a partially ordered set). A multivalued mapping $f$ on $X$ is called *strictly contracting*, respectively, *contracting* if, for all $x, y \in X$ with $x \neq y$ and for every $a \in f(x)$, there exists an element $b \in f(y)$ such that $d(a, b) < d(x, y)$, respectively, $d(a, b) \leq d(x, y)$.

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

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

The following theorem was proved in [14]. 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 $\Gamma$.

**4.7 Theorem** Let $(X, d, \Gamma)$ be a spherically complete ultrametric space. Let $f : X \to 2^X$ be a non-empty contraction which is strictly contracting on orbits. Moreover, assume that for every $x \in X$, Min$\Pi_x$ is finite and that every element of $\Pi_x$ has a lower bound in $\mathrm{Min}_x$. Then $f$ has a fixed point.

### *4.2.3 Quasi-Metric Spaces*

Just as in the case of single-valued mappings, one can establish fixed-point theorems for multivalued mappings using quasi-metrics. In fact, we present several such in this section, all of which are new and all of which are under investigation in relation to applications to the semantics of disjunctive programs and databases.

**4.8 Definition** Let $(X, d)$ be a quasi-metric space. A multivalued mapping $f : X \to 2^X$ is called a *contraction* if there exists some $0 \leq k < 1$ such that, for all $x, y \in X$ and for all $a \in f(x)$, there exists $b \in f(y)$ satisfying $d(a, b) \leq kd(x, y)$.

A multivalued mapping $f$ on $X$ is called *non-expanding* if, for all $x, y \in X$ and for all $a \in f(x)$, there exists $b \in f(y)$ satisfying $d(a, b) \leq d(x, y)$.

A multivalued mapping $f$ on $X$ is called *CS-continuous* if, for every (forward) Cauchy sequence $(x_n)$ in $X$ with limit $x$ and for every choice of $y_n \in f(x_n)$, we have that $(y_n)$ is a (forward) Cauchy sequence and $\lim y_n \in f(x)$.

**4.9 Theorem** Let $(X, d)$ be a CS-complete quasi-metric space and let $f : X \to 2^X$ be a multivalued contraction which is CS-continuous and non-empty (i.e. for all $x \in X$, $f(x) \neq \emptyset$). Then $f$ has a fixed point.

**Proof:** The point $x_\omega$ is obtained exactly as in the proof of Theorem 4.5. By CS-continuity of $f$, it follows that $x_\omega$ is a fixed point of $f$. ■

**4.10 Theorem** Let $(X, d)$ be a CS-complete quasi-metric space and let $f : X \to 2^X$ be a non-expanding multivalued mapping which is CS-continuous and non-empty. Suppose that there exists $x_0 \in X$ and $y \in f(x_0)$ with $x_0 \leq_d y$, i.e. such that $d(x_0, y) = 0$. Then $f$ has a fixed point.

**Proof:** Choose $x_1 \in f(x_0)$ such that $d(x_0, x_1) = 0$. Since $f$ is non-expanding, there is some $x_2 \in f(x_1)$ with $d(x_1, x_2) \leq d(x_0, x_1) = 0$. Inductively, we obtain a sequence $(x_n)$ with $d(x_n, x_{n+k}) \leq \sum_{i=0}^{k-1} d(x_{n+i}, x_{n+i+1}) = 0$. Hence, $(x_n)$ is a Cauchy sequence and has a limit $x_\omega$. By CS-continuity of $f$, $x_\omega$ is a fixed point of $f$. ■

We can weaken the assumptions of the previous results and introduce the following notions.

**4.11 Definition** Let $f$ be a multivalued mapping. A sequence $(x_n)$ in $X$ such that $x_{n+1} \in f(x_n)$ for all $n \in \mathbb{N}$ is called an *$\omega$-orbit* of $f$. For a multivalued mapping $f$ defined on a quasi-metric space, we call $f$ *orbitally CS-continuous* if, for every orbit $(x_n)$ of $f$ which is a Cauchy sequence, we have $\lim x_n \in f(\lim x_n)$.

Both the previous theorems hold if the condition on $f$ being CS-continuous is relaxed to being orbitally CS-continuous.

Next, by virtue of the duality between quasi-metrics and partial orders, we carry over the previous result to partial orders.

**4.12 Proposition** Let $(D, \leq)$ be a partial order such that $(D, d_{\leq})$ is a CS-complete quasi-metric space. Then the following hold.

(i) $(x_n)$ is a Cauchy sequence if and only if it is eventually increasing.

(ii) Every increasing sequence has a least upper bound and the sequence converges to this least upper bound.

(iii) If $(D, \leq)$ has a least element, then it is an $\omega$-cpo.

**4.13 Proposition** Let $(D, \leq)$ be an $\omega$-cpo, let $f$ be a multivalued mapping on $D$ and consider $d_{\leq}$. Then the following hold.

(i) $f$ is non-expanding if and only if $f$ is *monotonic* in the sense that whenever $x \leq y$ and $a \in f(x)$, there exists $b \in f(y)$ with $a \leq b$.

(ii) $f$ is orbitally CS-continuous if and only if for every $\omega$-orbit of $f$ which is eventually increasing with least upper bound $x$, we have $x \in f(x)$.

By the previous two propositions, we now easily obtain our final result, which is another generalization of the Knaster-Tarski Theorem 2.4.

**4.14 Theorem** Let $(D, \leq)$ be an $\omega$-cpo and let $f$ be a non-empty multivalued mapping on $D$ which is monotonic, and which has the property that every $\omega$-orbit of $f$ which is eventually increasing with least upper bound $x$ satisfies $x \in f(x)$. Then $f$ has a fixed point.

We remark, again, that all these theorems have potential applications to semantics which are under investigation. In addition, we have other results in which we employ a notion of "generalized metric" which takes its values in anordered Abelian semigroup, as in [10], but these will be discussed elsewhere.

# 5 Conclusions

Multivalued mappings and their fixed points arise naturally in the context of the semantics of disjunctive programs and databases. Therefore, there is a need for methods and techniques which can provide these fixed points. Of course, one expects such methods to relate to results already known for single-valued mappings, and indeed we have shown that this is the case. However, due to the rather special nature of the multivalued mappings we encountered, namely those arising from disjunctive programs $\Pi$, it was possible to relate fixed points of the multivalued mapping $T_{\Pi}$ to fixed points of the single-valued mappings $T_P$ associated with the normal derivatives $P$ of $\Pi$.

Many questions are raised by this work. One such, for example, is that of characterizing the normal derivatives which determine any one of the standard models of $\Pi$. Another is the examination of syntactic conditions which enable one to easily apply the fixed-point theorems we have discussed, which are mostly stated as purely mathematical results; this is a question which we have only lightly touched on here. A third is the question of which of our theorems can be given constructive proofs. Other questions can be posed, and indeed all are under investigation and will be considered elsewhere. Thus, we regard this paper as a continuation of a programme of work, begun in [17, 18], in which semantical questions within declarative programming are investigated using the tools of domain theory.

# References

[1] Apt KR, Pedreschi D. Reasoning about Termination of Pure Prolog Programs. Information and Computation 1993; 106:109-157

[2] Dugundji J, Granas A. Fixed Point Theory. Polish Scientific Publishers, Warsaw, 1982

[3] Fitting M. A Kripke-Kleene Semantics for Logic Programs. Journal of Logic Programming 1985; 4:295-312

[4] Fitting M. Metric Methods: Three Examples and a Theorem. Journal of Logic Programming 1994; 21(3):113-127

[5] Gelfond M, Lifschitz V. The Stable Model Semantics for Logic Programming. In: Kowalski RA, Bowen KA (eds) Proceedings of the Fifth International Conference and Symposium on Logic Programming. MIT Press, 1988, pp 1070-1080

[6] Gelfond M, Lifschitz V. Classical Negation in Logic Programs and Disjunctive Databases. New Generation Computing 1991; 9:365-385

[7] Hitzler P. Topology and Logic Programming Semantics. Diplomarbeit, University of Tübingen, 1998

[8] Hitzler P, Seda AK. Simplifying Termination Proofs for Pure Prolog Programs. Preprint, 1999, pp 1-18

[9] Jachymski J. A Contribution to Fixed-Point Theory in Quasi-Metric Spaces. Publ. Math. Debrecen 1993; 43(3-4):283-288

[10] Khamsi MA, Kreinovich V, Misane D. A New Method of Proving the Existence of Answer Sets for Disjunctive Logic Programs: A Metric Fixed Point Theorem for Multivalued Maps. Proc. of the Workshop on Logic Programming with Incomplete Information. Vancouver, British-Columbia, Canada, 1993, pp 58-73

[11] Khamsi MA, Misane D. Disjunctive Signed Logic Programs. Fundamenta Informaticae 1997/98; 32(3-4):349-357

[12] Lloyd JW. Foundations of Logic Programming, Second Edition. Springer-Verlag, Berlin, 1988.

[13] Prieß-Crampe S. Der Banachsche Fixpunktsatz für Ultrametrische Räume. Results in Mathematics 1990; 18:178-186

[14] Prieß-Crampe S, Ribenboim P. Ultrametric Spaces and Logic Programming. Preprint, October 22, 1997

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

[16] Rutten JJMM. Elements of Generalized Ultrametric Domain Theory. Theoretical Computer Science 1996; 170:349-381

[17] Seda AK. Topology and the Semantics of Logic Programs. Fundamenta Informaticae 1995; 24(4):359-386

[18] Seda AK. Quasi-metrics and the Semantics of Logic Programs. Fundamenta Informaticae 1997; 29(2):97-117

[19] Seda AK, Hitzler P. Topology and Iterates in Computational Logic. Proceedings of the 12th Summer Conference on Topology and its Applications: Special Session on Topology in Computer Science, Ontario, August 1997. Topology Proceedings 1999; 22:427-469

[20] Seda AK, Hitzler P. Strictly Level-Decreasing Logic Programs. In: Butterfield A, Flynn S (eds) Proceedings of the Second Irish Workshop on Formal Methods (IWFM'98). Electronic Workshops in Computing, British Computer Society, 1999, pp 1-18

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

[22] Stoltenberg-Hansen V, Lindström I, Griffor E. Mathematical Theory of Domains. Cambridge University Press, Cambridge, 1994