Fixed Point Theorems in Logic Programming

Mohamed A. Khamsi[1] and Driss Misane[2]
[1] Department of Mathematical Sciences,
University of Texas at El Paso, El Paso, TX 79968.
[2]Departement de Mathematiques,
Universite Mohamed V, Faculte des Sciences, Rabat, Morocco.
email:[1] mohamed@math.utep.edu
URL:[1] http://www.mdrkhamsi.com

1 Introduction

Very often scientific branches which were thought to be completely disparate are suddenly seen to be related. This is the case for example with mathematics of which the level of sophistication applied to various sciences has changed drastically in recent years. Fixed point theory furnishes good example of a central concept with multitudes of different uses. It has always been a major theoretical tool in fields as widely apart as differential equations, topology, economics, game theory, dynamical systems (and chaos), optimal control, functional analysis, logic programming and artificial intelligence. Moreover, more or less recently, the usefulness of the concept for applications increased enormously by the development of accurate and efficient techniques for computing fixed points, making fixed point methods a major weapon in the arsenal of the applied mathematician.

2 Elementary fixed point theorems

The theory of fixed points is concerned with the conditions which guarantee that a map $F : X \to X$ of a set X into itself admits one or more fixed points, that is, points $x \in X$ for which $F(x) = x$. The set of fixed points of F will be denoted $Fix(F)$.
In the sequel, we will discuss two majors fixed point theorems which are based on a notion of completeness. Although the spaces involved are of different nature, there is a similarity between the two theorems.

2.1 Knaster-Tarski theorem [8]

Let $(P, \leq)$ be a partially ordered set and $M \subset P$ a non-empty subset. Recall that an upper (lower) bound for M is an element $p \in P$ with $m \leq p$ ($p \leq m$) for each $m \in M$; the least-upper (greatest-lower) bound of M will be denoted $\sup M$ (resp. $\inf M$). Of course there is, in general, no reason for $\inf M$ and $\sup M$ to exist. $(P, \leq)$ will be called complete if for every subset $M \subset P$, $\inf M$ and $\sup M$ do exist. Recall that $M \subset P$ is said to be linearly ordered if for every $m_1, m_2 \in M$ we have $m_1 \leq m_2$ or $m_2 \leq m_1$. A linearly ordered subset of P is called a chain. A map $F : P \to P$ is monotone (also called isotone or increasing) if $F(x) \leq F(y)$ whenever $x \leq y$.

Theorem of Knaster-Tarski *Let $(P, \leq)$ be a complete ordered set and $F : P \to P$ monotone. Then $Fix(F)$ is not empty, that is F has a fixed point. Moreover, $Fix(F)$ is a complete ordered subset of P. In particular, it has the least and greatest elements.*

Note that one classical proof is based on a transfinite iteration of F. Indeed, denote m to be the least element of P. Clearly, we have $m \leq F(m)$. Since F is monotone, we will get $F^n(m) \leq F^{n+1}(m)$ for every $n = 1, 2, ...$ Set $x_n = F^n(m)$. Since P is complete then $x_\omega = \sup_n x_n$ does exist. There is no reason for x_ω to be a fixed point. Therefore, we should continue the process, that is if $(x_\alpha)_{\alpha<\beta}$ are constructed, then $x_\beta = \sup_{\alpha<\beta} x_\alpha$ if β is a limit ordinal otherwise $x_\beta = F(x_{\beta-1})$. The (x_α) defines a chain which will eventually stop, that is there exists an ordinal α_0 such that $x_{\alpha_0} = x_{\alpha_0+1} = F(x_{\alpha_0})$. It is very easy to see that x_{α_0} is the least fixed point of F. By iterating F over the greatest element of P, one will generate the greatest fixed point of F. A natural question arises about what do we know of the ordinal α_0. One easy answer has to do with the size of the set P. Not very much encouraging since P can be very big. Another partial answer is given by what we call continuity. Indeed, the map $F : P \to P$ is called continuous if for every subset $M \subset P$, we have $\sup F(M) = F(\sup M)$. Note that if F is monotone, then $\sup F(M) \leq F(\sup M)$ but not the equality. It is well known that if $F : P \to P$ is continuous and monotone, then $\alpha_0 = \omega$ regardless of the given point $x \in P$ which initiates the iteration process, where ω is the first countable ordinal. Indeed, fix $x \in P$ and consider the iteration sequence $(F^n(x))$. Then we have

$$F(x_\omega) = F(\sup_n F^n(x)) = \sup_n F^{n+1}(x) = x_\omega$$

which shows that the fixed point will always be reached after ω-iteration.

Since whenever negation is involved in logic programs, it is to be expected that the mappings involved will no longer be monotone. As a matter of fact, we may have an anti-monotonic behavior. Recall that a map $F : P \to P$, where $(P, \leq)$ is a partially ordered set, is said to be anti-monotonic or decreasing if for every $m_1, m_2 \in P$ we have $F(m_2) \leq F(m_1)$ whenever $m_1 \leq m_2$. Clearly, $F^2 = FoF$ will be monotone. Therefore, F^2 will have fixed point (called periodic point of F of period 2). Let m be the lowest fixed point of F^2 and M be the greatest fixed point of F^2. Then we have $F(m) = M$ and $F(M) = m$.

2.2 Banach contraction principle

The theorem of Banach is the simplest and one of the most versatile results in fixed point theory. Being based on an iteration process, it can be implemented on a computer to find the fixed point of a contractive map producing approximations of any required accuracy and, moreover, even the number of iterations needed to get specified accuracy can be determined. First, recall that a pair (M, d) is called a metric space [5] if the map $d : M \times M \to [0, +\infty)$ satisfies

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

(ii) $d(x, y) = d(y, x)$

(i) (Triangle inequality) $d(x, y) \leq d(x, z) + d(z, y)$

for every $x, y, z \in M$. d is called a distance.

Using the distance, we can give a meaning to the notion of closeness. Indeed we will say x is close to y if and only if $d(x, y)$ is small. We will say that a metric is complete if whenever a sequence (x_n) of points from M is such that x_n and x_m are close for n and m big enough (such sequence is called a Cauchy sequence), then there exists a point $x \in M$ such that x_n gets close to x when n is big. In a more correct fashion, we will say that M is complete if for every sequence (x_n) such that $\lim d(x_n, x_m) = 0$ when n, m go to infinity, then there exists $x \in M$ such that $\lim d(x_n, x) = 0$ when n goes to infinity. Note that not all metric spaces are complete. A mapping $F : M \to M$ is called Lipschitzian if there exists a constant L such that $d(F(x), F(y)) \leq Ld(x, y)$ for every $x, y \in M$; the smallest such constant L is called the Lipschitz constant $Lip(F)$ of F. If $Lip(F) < 1$, the map F is called contractive.

Theorem (Banach contraction principle) [5] *Let (M, d) be a complete metric space and $F : M \to M$ a contractive mapping. Then F has a unique fixed point w, that is $Fix(F) = \{w\}$. Moreover, for any $x \in M$, we have*

$$d(F^n(x), w) \leq \frac{L^n}{1 - L} d(x, F(x) \ ; \quad \textit{for } n = 1, 2, ..$$

where $L = Lip(F) < 1$. Therefore we have $\lim F^n(x) = w$ when $n \to \infty$.

It is clear from the theorem that the error of approximating w by the nth iteration when starting from a given $x \in M$ is completely determined by the contraction constant $Lip(F)$ and the initial displacement $d(x, F(x))$.

It is worth to mention here that the theorem of Banach contains two important conclusions. The first one deals with the uniqueness of the fixed point. The other one deals with the way we reach this fixed point since the iteration reaches the fixed point after ω-iteration, where ω is the first countable ordinal.

Remark. Before, we get to the study of multivalued mappings, we would like to make an elementary introduction to generalized metric spaces. The main idea behind this new structure is to replace the set of positive numbers as the set value for the distance by something more general. The reason behind is that many structures are not of continuous nature. More precisely sets which are of discrete nature will have hard time to fit into the class of classical metric spaces. Therefore people working in this area were the pioneers in developing the appropriate extension. Let us describe what they did. Let $\mathcal{V}$ be a set with a binary operation which will be denoted $\oplus$. We will assume that $\oplus$ enjoys most of the properties that the classical addition does. In particular, we assume that there exists a zero element $0 \in \mathcal{V}$ which satisfies $u \oplus 0 = 0 \oplus u = u$ for every $u \in \mathcal{V}$. We will also assume (as for the set of positive numbers) that we have an order $\leq$ such that $0 \leq u$ and $u \oplus v \leq u' \oplus v'$ provided $u \leq u'$ and $v \leq v'$ for every $u, v, u', v' \in \mathcal{V}$. Note that the set $\mathcal{V}$ is not behaving totally like the set of positive numbers since we do not have a priori a multiplication operation defined on $\mathcal{V}$.

Definition. Let $\mathcal{V}$ be a set as described above and M be an arbitrary set. The mapping $d : M \times M \to \mathcal{V}$ is called a generalized distance if

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

(ii) $d(x, y) = d(y, x)$

(i) (Triangle inequality) $d(x, y) \leq d(x, z) \oplus d(z, y)$

for every $x, y, z \in M$. A pair (M, d) is called a generalized metric space.
We should mention that in the original definition of generalized distance, the condition (ii) is replaced by

$$d(y, x) = \tau(d(x, y))$$

where τ is an involution, that is $\tau(\tau(x)) = x$.

Example of $\mathcal{V}$ Because of applications to disjunctive logic programming, one interesting example of the abstract set $\mathcal{V}$ is given by the expressions $2^{-\alpha}$, where α is a countable ordinal, augmented with the element 0. An order on $\mathcal{V}$ is defined as follows: $0 \leq v$, $\forall v \in V$, and $2^{-\alpha} \leq 2^{-\beta}$ whenever $\beta \leq \alpha$. The operation $\oplus$ is defined by $u \oplus v = \max(u, v)$. Note that $\mathcal{V}$ enjoys a natural multiplication property defined as: $2^{-\alpha}.2^{-\beta} = 2^{-(\alpha+\beta)}$.

2.3 The multivalued case

Multivalued mappings play a major role in studying disjunctive logic programs. First, recall that a map $F : X \to 2^X$ is called a multivalued map. A singlevalued map is a multivalued map F such that $F(x)$ is a singleton. A point $x \in X$ is a called a fixed point of a multivalued map F if $x \in F(x)$ holds. Although the fixed point theory for singlevalued maps is very rich and well developed, the multivalued case is not. Few theorems are known in topological spaces but nothing major with direct impact on the study of disjunctive logic programs. For example, it is unknown to us whether a multivalued version of Knaster-Tarski theorem is known. In what follows we will discuss this case.
Let $(P, \leq)$ be a partially ordered set. Define the relation $\prec_r$ (r for restriction) in 2^P by

$$A \prec_r B \Longleftrightarrow \forall y \in B \;\; \exists x \in A \quad x \leq y$$

it is clear that $\prec_r$ is a preorder but not an order ($\prec_r$ is not antisymetric). We can also define the extension preorder $\prec_e$ in 2^P by

$$A \prec_e B \Longleftrightarrow \forall x \in A \;\; \exists y \in B \quad x \leq y \; .$$

We say that a multivalued mapping $T : P \to 2^P$ is $\prec_r$-monotone if

$$x \leq y \Longrightarrow T(x) \prec_r T(y)$$

Remark: Recall that a subset $M \subset P$ is called antichain if for every $m_1, m_2 \in M$, we do not have $m_1 \leq m_2$ neither $m_2 \leq m_1$. Denote

$$M = \{A \in 2^P ; A \text{ is an antichain } \}$$

then $(M, \prec_r)$ and $(M, \prec_e)$ are ordered sets because $\prec_r$ and $\prec_e$ are orders in M. Since the study of stable model semantics for disjunctive logic programs involves the preorder $\prec_r$, we will not discuss the preorder $\prec_e$.

Definition. Let P be a partially ordered set and $T : P \to 2^P$ a multivalued mapping. We say that the family (x_β), where β runs through the ordinals, is a decreasing T-orbit if

$$\begin{cases} x_{\beta+1} \in T(x_\beta) \\ x_{\beta+1} \leq x_\beta \end{cases}$$

Theorem (Knaster-Tarski) *Let P be a complete ordered set and T be an $\prec_r$-increasing multivalued mapping from P into 2^P such that for every $x \in P$,*

1. *$T(x)$ is not empty (i.e. $T(x) \neq \emptyset$)*
2. *For every decreasing T-orbit (x_β), there exists $x \in P$ such that $x \in T(\inf x_\beta)$ and $x \leq x_\beta$ for all β,*

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

Remark: The second hypothesis of the theorem holds if $T(x)$ is finite for all $x \in P$.

Let us turn our attention to Banach contraction principle for multivalued mappings. First we should mention that after the following theorem was proved, we discovered that many multivalued versions of the theorem of Banach do exist.

Theorem (Banach contraction principle) *Let (M, d) be a complete metric space. Let $F : M \to 2^M$ be a multivalued map satisfying for every $x, y \in M$*

$$\forall x_1 \in T(x)\ ;\ \exists y_1 \in T(y) \text{ such that } d(x_1, y_1) \leq Ld(x, y)$$

where $L < 1$. Then F has a fixed point provided that $F(x)$ is closed and nonempty for every $x \in M$.

A similar result for multivalued mappings defined on a generalized metric space holds. Indeed, Let (M, d) be a generalized metric space with set-values for d to be the set $\mathcal{V}$ described in the example above. Then a multivalued map $F : M \to 2^M$ will be called 1/2-contraction if for every $x, y \in M$, and for every $z \in F(x)$, there exists $w \in F(y)$ such that

$d(z,w) \leq 1/2 d(x,y)$. In order to state an analogue to Banach contraction principle for generalized metric spaces, one would need to define the notion of completeness. We will not give the definition here but assure the reader that this definition is similar to the one described above.

Theorem *Assume that (M,d) is a complete generalized metric space and F a multivalued mapping defined on M which is $1/2$-contraction. Then F has a fixed point provided that $F(x)$ is nonempty and complete for every $x \in M$.*

3 Application to logic programming

A precise meaning or semantics must be associated with any logic program or a deductive database in order to provide its declarative specification, in a manner, which is independent of procedural considerations, context-free, and easy to manipulate, exchange and reason about. The problem of finding a suitable declarative or intended semantics is one of the most important and difficult problems in the theory of logic programming and deductive database. In this work, we will not discuss the merit of any of the semantics. Our goal is to discuss how fixed point theorems old or new can be used to answer some open questions regardless of the semantics used. For the sake of illustrating our point, we will consider the answer set semantics [4].

Let Lit be the set of ground literals in a first-order language $\mathcal{L}$. A rule r is an expression of the following form:

$$l_0 \mid l_1 \mid \cdots \mid l_n \leftarrow l_{n+1}, \cdots, l_m, not\ l_{m+1}, \cdots, not\ l_k$$

where $l_i \in Lit$. Set

$$Head(r) = \{l_0, l_1, \cdots, l_n\} \ \ Pos(r) = \{l_{n+1}, \cdots, l_m\} \ \text{ and } \ Neg(r) = \{l_{m+1}, \cdots, l_k\}$$

The rule r is said to be disjunctive if $n > 1$, i.e. $Head(r)$ has more than one element and nondisjunctive otherwise. An extended (disjunctive) logic program Π is a set of (disjunctive) rules. Instead of extended logic program, we will shorten it to program. In order to define the answer set semantics of extended logic programs, let us first consider programs without *not*.

Definition. Let Π be a program (disjunctive or not) for which $Neg(r)$ is empty for every $r \in \Pi$. A subset S of Lit, i.e $S \in 2^{Lit}$, is said to be closed with respect to Π if

for every $r \in \Pi$ such that $Pos(r) \subset S$, we have $Head(r) \cap S$ is not empty.

The set $S \in 2^{Lit}$ is an answer set of Π

1. If S contains complementary literals, then $S = Lit$.

2. S does not contain strictly a closed subset, i.e. if $A \subset S$ and A is closed with respect to Π then $A = S$.

The set of answer sets of Π is denoted by $\alpha(\Pi)$. If Π is not disjunctive, then $\alpha(\Pi)$ is a singleton, i.e. Π has one answer set and if it is disjunctive then $\alpha(\Pi)$ may contain more than one element.

Now let Π be a program that may contain *not* (general case). For $S \in 2^{Lit}$, consider the program Π^S defined by the set of rules

1. If $r \in \Pi$ such that $Neg(r) \cap S$ is not empty, then $r \notin \Pi^S$
2. If $r \in \Pi$ such that $Neg(r) \cap S$ is empty, then the rule r' defined by $Head(r') = Head(r)$, $Pos(r') = Pos(r)$ and $Neg(r') = \emptyset$, belongs to Π^S.

Clearly the program Π^S does not contain *not*.

Definition.

1. The set $S \in 2^{Lit}$ is said to be an answer set of Π if $S \in \alpha(\Pi^S)$.
2. Define the GL-operator $T : 2^{Lit} \to 2^{Lit}$ by

$$T(X) = \alpha(\Pi^X) \ .$$

Using the GL-operator T, the set S is an answer set of Π iff $S \in T(S)$. If Π is not disjunctive then S is an answer of Π iff $S = T(S)$. In both cases, S is a fixed point of T.

Main Problem. Find conditions on Π that will insure the existence of an answer set, i.e. a fixed point of T.

In what follows, we discuss a basic property of the GL-operator.

Property. Let Π be a program. Then if $X \subset Y$, $X, Y \in 2^{Lit}$, then

$$\forall A \in \alpha(\Pi^X) \;\; \exists B \in \alpha(\Pi^Y) \;\text{ such that }\; B \subset A$$

i.e. $\alpha(\Pi^Y) \prec_r \alpha(\Pi^X)$.

Remark. When the program Π is not disjunctive, then we have

$$X \subset Y \implies \alpha(\Pi^Y) \subset \alpha(\Pi^X)$$

in another word the GL-operator T is anti-monotonic. Therefore, T^2 is monotonic. Using the above discussion, we know that there exist a least fixed point $lfp(T^2)$ and a greatest fixed point $gfp(T^2)$ of T^2. We also know that $T(lfp(T^2)) = gfp(T^2)$ and $T(gfp(T^2)) = lfp(T^2)$. In another world, the set $\{lfp(T^2), gfp(T^2)\}$ is stable under the action of T. Using this result, Baral [1] defined the notion of stable class semantics. It is worth to mention that there are logic programs which fail to possess stable models while stable classes always exist. Therefore, for these programs we will not be able to consider the stable semantics. Baral was also successful to connect stable class semantics to well-founded semantics. Indeed, he proved that if Π is a normal logic program, the well-founded semantics is characterized by a particular stable class $\mathcal{C}$, i.e. a ground atom is true (resp false) in the well-founded semantics of Π iff A is true (resp. false) in all interpretations in $\mathcal{C}$. Moreover, $\mathcal{C} = \{lfp(T^2), gfp(T^2)\}$. It is amazing that the Gelfond-Lifschitz operator could catch the well-founded semantics in this way. Thereofre, a natural question to be asked is whether a similar result holds for disjunctive logic programs. The answer is still unknown. The difficulty resides in considering T^2 since T is multivalued and getting a monotonic mapping. In another world, we do not know whether a stable classes do exist for disjunctive logic programs.

3.1 Signed disjunctive programs

The notion of a signing for a program was introduced by Kunen [6] (defined on the predicate dependency graph), who used it as a tool in his proof that two-valued and three valued completion semantics coincide on the class of strict normal programs. Gelfond and Lifschitz recasted this notion and redefined it directly on the rules of programs. Recently Turner [9] did study extensively signed programs. In particular, he was able to give an alternative proof of Fages'result [2] using signed programs.

Definition. We will say that Π is **signed** if there exists $S \in 2^{Lit}$, called a signing, such that for every $r \in \Pi$ we have

1. if $Neg(r) \cap S$ is empty, then $Head(r) \subset S$ and $Pos(r) \subset S$. Let Π_s be the program generated by these rules.

2. If $Neg(r) \cap S$ is not empty, then $Head(r) \cap S = \emptyset$, $Pos(r) \cap S = \emptyset$ and $Neg(r) \subset S$. Let $\Pi_{\bar{s}}$ be the program obtained from these rules, where $\bar{S}$ denotes the complement of S, i.e. $\bar{S} = Lit - S$.

Clearly the two subprograms Π_s and $\Pi_{\bar{s}}$ are disjoint and $\Pi = \Pi_s \cup \Pi_{\bar{s}}$.

For nondisjunctive signed programs with positive head, Gelfond and Lifschitz [4] proved the existence of a consistent answer set. It is still unknown if disjunctive signed programs have answer sets. For a more restrictive class of programs called semi-disjunctive, we have a positive answer.

Definition. A signed program Π is said to be **semi-disjunctive** if there exists a signing S such that Π_s is nondisjunctive.

Note that what Turner defines as signed disjunctive programs is what we call semi-disjunctive. One of the reason why Turner was interested into such programs is to prove the existence of answer sets for the two guns domain example which is a variant of the Yale

Shooting domain:

Example: Two guns domain. The story is about a pilgrim and a turkey. The pilgrim has two guns. Initially, the turkey is alive, but if the pilgrim fires a loaded gun, the turkey dies. Furthermore, at least one of the two guns is loaded initially. This clearly implies that the turkey will be dead if the pilgrim performs any of the following sequences of actions :

1. wait, shoot gun one, shoot gun two
2. wait, shoot gun two, shoot gun one.

The following program Π formalizes the two-gun domain.

1. $Holds(Alive, S_0) \leftarrow$
2. $Holds(Loaded_1, S_0) \mid Holds(Loaded_1, S_0) \leftarrow$
3. $\neg Holds(Alive, Result(Shoot_1, s)) \leftarrow Holds(Loaded_1, s)$
4. $Noinertial(Alive, Shoot_1, s) \leftarrow not \neg Holds(Loaded_1, s)$
5. $\neg Holds(Alive, Result(Shoot_2, s)) \leftarrow Holds(Loaded_2, s)$
6. $Noinertial(Alive, Shoot_2, s) \leftarrow not \neg Holds(Loaded_2, s)$
7. $Holds(f, Result(a, s)) \leftarrow Holds(f, s),\ not Noinertial(f, a, s)$
8. $\neg Holds(f, Result(a, s) \leftarrow \neg Holds(f, s),\ not Noinertial(f, a, s)$
9. $Holds(f, S_0) \mid \neg Holds(f, S_0) \leftarrow$

This program is signed semi-disjunctive with $S = \{Noinertial(f, a, s)\}$ as a signing. It is not hard to generalize this program to more than two guns and still have a signed semi-disjunctive program.

Note that the main result of [9] states that signed semi-disjunctive programs have a consistent answer set provided they have at least one head-consistent cover. One of the condition ensuring the existence of a head-consistent cover is to assume that the head of the program is consistent. For example, the above program is head-consistent. We weakened this assumption into the notion of safe programs.

Definition. A disjunctive program Π is said to be safe with respect to a partition (Π_1, Π_2) if for every $Y \in 2^{Lit}$ and $X \in \alpha(\Pi_1^Y)$, X does not activate two contrary rules.

Recall that the pair (Π_1, Π_2) defines a partition for the program Π if Π_1 and Π_2 are two disjoint subprograms of Π such that $\Pi = \Pi_1 \cup \Pi_2$, and that two rules r_1 and r_2 are said to be contrary if there exists a literal $l \in Lit$ such that $l \in Head(r_1)$ and $\neg l \in Head(r_2)$.

An example of a safe program (which is not head-consistent) is the program formalizing the classical flying birds story:

Exemple. Suppose that we are told that penguins are birds that do not fly, that birds normally fly, and that Tweety is a bird and not a penguin and Sam is a penguin. Let us also assume that this information is complete. Therefore, we can represent knowledge from the exemple by the logic program Π consisting of the rules:

1. $f(X) \leftarrow b(X),\ not\ ab(f,b,X)$
2. $b(X) \leftarrow p(X)$
3. $ab(f,b,X) \leftarrow p(X)$
4. $\neg f(X) \leftarrow p(X)$
5. $\neg f(X) \leftarrow \neg b(X)$
6. $b(t) \leftarrow$
7. $p(s) \leftarrow$
8. $\neg p(t) \leftarrow$

Note that t (for Tweety) and s (for Sam) are the only constants allowed by the program. Consider the two subprograms:

$$\Pi_1 \begin{cases} ab(f,b,X) \leftarrow p(X) \\ p(s) \leftarrow \end{cases}$$

$$\Pi_2 \begin{cases} f(X) \leftarrow b(X),\ not\ ab(f,b,X) \\ b(X) \leftarrow p(X) \\ \neg f(X) \leftarrow p(X) \\ \neg f(X) \leftarrow \neg b(X) \\ b(t) \leftarrow \\ \neg p(t) \leftarrow \end{cases}$$

It is clear that (Π_1, Π_2) forms a partition for Π for which it is safe.

Using the multivalued-Tarski fixed point theorem, we prove the following result:

Theorem. Let Π be a signed safe semi-disjunctive program. Then Π has a consistent answer set.

The assumption semi-disjunctive is not necessary (for more on this question see [11]).

Remark. In order to see how the ideas behind the proof of the multivalued-Tarski theorem work, let us consider the following program:

$$\Pi \begin{cases} c \mid d & \leftarrow & not\ a \\ a \mid b & \leftarrow & not\ c \\ a & \leftarrow & not\ d \\ b & \leftarrow & \end{cases}$$

A signing for this program is $S = \{a, b\}$ (hence $\bar{S} = \{c, d\}$). Consider the operator $T : 2^{\bar{S}} \to 2^{2^{\bar{S}}}$ defined by

$$T(X) = \alpha(\Pi_{\bar{s}}^{\alpha(\Pi_s^X)})$$

Set $Z_0 = \{c, d\}$. Then we have

$$T\Big(\{c, d\}\Big) = \Big\{\{c\}, \{d\}\Big\}$$

We have a choice for Z_1.

1. If $Z_1 = \{c\}$, then $T(Z_1) = \{\emptyset\}$. And since $T\Big(\{\emptyset\}\Big) = \{\emptyset\}$, we get $X_2 = \{\emptyset\}$. Hence $X_1 = \alpha(\Pi_s^{X_2}) = \{a, b\}$. Therefore an answer set for Π is $X_1 \cup X_2 = \{a, b\}$.

2. If $Z_1 = \{d\}$, then $T(Z_1) = \Big\{\{c\}, \{d\}\Big\}$. Therefore $\{d\}$ is a fixpoint of T which implies that $X_2 = \{d\}$. Hence $X_1 = \alpha(\Pi_s^{X_2}) = \{b\}$. Therefore an answer set for Π is $X_1 \cup X_2 = \{d, b\}$.

As one can see the iterations gave the two only answer sets of Π. Note that whenever the program is finite, the associated iterations of T will stop after a finite number of steps.

3.2 Locally stratified programs

Fitting [3] was the first to use Banach contraction principle to prove the existence of stable models. We should mention that, since one of the conclusions of the theorem of Banach is the uniqueness of the fixed point, its possible use as a tool to prove the existence of models will work if the appropriate class of programs do have one model. This is the case for stratified and more generally locally stratified. Since Fitting's approach is based on level mappings, a more general class of programs for which such mappings do exist can be considered. When the set of literals *Lit*, associated to a given program, is countable then the classical form of the theorem of Banach works. But we know that for example if a program is locally stratified but not countably stratified, then the size of the program is bigger than ω, the first countable ordinal. Therefore, one can not use the theorem of Banach in its classical form and will need to use generalized metric spaces instead. Moreover, if the program is disjunctive, then we should combine a mutlivalued version of Banach contraction principle in generalized metric spaces. We see how these extensions are helpful to better understand disjunctive programs. Although the following theorem is stated in terms of locally stratified disjunctive programs, it is very easy to adapt it to the class of programs considered by Fitting (using

level mappings).
Let us recall the definition of a stratified logic program. Let Π be a program and Lit the associated set of literals. The program Π is said to be locally stratified if

$$Lit = \bigcup_{\alpha \in \Gamma} Lit_\alpha$$

where Γ is the set of countable ordinals and the Lit_α are disjoint sets, such that for every rule $r \in \Pi$, we have

(i) if $l \in Head(r)$ and $l' \in Pos(r)$ with $l \in Lit_\alpha$ and $l' \in Lit_\beta$, then $\alpha \geq \beta$;

(ii) if $l \in Head(r)$ and $l' \in Neg(r)$ with $l \in Lit_\alpha$ and $l' \in Lit_\beta$, then $\alpha > \beta$.

A decomposition $\{Lit_\alpha\}$ of Π satisfying the above conditions is called a stratification of Π.

Theorem. *Let Π be a stratified extended disjunctive logic program. Then Π has an answer set.*
Idea of the proof. Let Lit_α denote the stratas of Π. Define the generalized distance $D : 2^{Lit} \times 2^{Lit} \to \mathcal{V}$ as follows:

(1) if $A = B$, then $d(A, B) = 0$;

(2) if $A \neq B$, then $d(A, B) = 2^{-\alpha}$, where α is the smallest ordinal for which $A \cup Lit_\alpha \neq B \cup Lit_\alpha$.

Consider the GL-operator $T(X) = \alpha(X)$. Then T satisfies the assumptions of Banach contraction principle. Therefore, T has a fixed point which happens to be an answer set to the program Π. For more on this theorem, the reader may consult [10].

Remark. Let us point out that the above ideas go beyond the class of stratified programs. For example, Fitting provides an example when metric fixed point theorems proves the existence of a stable set for a non-stratified program. Namely, he considers a program that describes wining positions in a positional game, in which players make moves in turn and there are no draws. If a position is winning for one of the players, then no further moves are possible. This program consists of the rules and the facts. Rules are of the following type:

$$win(X) \leftarrow move(X, Y), not\ win(Y),$$

and the facts (of the type $move(X, Y)$) describe possible moves. Predicate $win(X)$ means that X is a winning position, i.e., if a player is in a position X, then there exists a strategy that enables him to win (no matter what the actions of the opposite player are).

The (informal) meaning of the above rule is as follows: if a player is in a position X, and he can move into another position Y that is not winning (i.e., losing) for the opposite player, then he wins. If he cannot make such a move, this means that wherever he moves to, the resulting position is winning for the second player. So, if no such move exists, then X is a losing position for X.

Let's modify this example into an example where a metric fixed point theorem helps to find a stable model for a non-stratified disjunctive logic program. We will consider the same game, but this time, we will assume that the players are still training (it is not yet a championship). So, if a person is about to win, then instead of going all the way to his victory, he can stop the game and teach another player (i.e., explain how he could win). A program that describes this situations contains the same facts $move(X, Y)$, but slightly different rules:

$$can\,win(X) \leftarrow move(X, Y), not\ can\,win(Y)$$

$$win(X) \vee teach(X) \leftarrow can\,win(X).$$

Rules of the first type describe when a player can win. Rules of the second type tell that if a player can win, then he will either win, or teach. This program is non-stratified. However, for this program, the mapping $S \to \alpha(\Pi^S)$ is a contraction and therefore, the fixed point theorem for multi-valued contractions proves that this mapping has a fixed point (i.e., proves the existence of a stable model of Π).

References

[1] C.R. Baral, V.S. Subrahmanian, "Stable and Extension Class Theory for Logic Programs and Default Logics", Journal of Automated Reasoning, 8(1992), 345-366.

[2] F. Fages, "Consistency of Clark's completion and existence of stable models", Journal of Methods of Logic in Computer Science, 1(1994),51-60.

[3] M. Fitting, "Metric methods, three examples and a theorem", Journal of Logic Programming, 1993 (to appear).

[4] M. Gelfond, V. Lifschitz, "The stable model semantics for logic programming", R. Kowalski and K. Bowen (eds), Logic programming: Proceedings of the 5th Intl. Conference and Symposium, 1988, pp. 1070-1080.

[5] J. Kelley, "General Topology", Van Nostrand, Princeton, NJ, 1955.

[6] K. Kunen, "Signed data dependencies in logic programs", Journal of Logic Programming, 7(3)(1989), 231-245.

[7] J.W. Loyd, "Foundations of logic programming", Springer-Verlag, N.J., 1987.

[8] A. Tarski, "A lattice-theoretical fixpoint theorem and its applications", Pacific Journal of Math., 5(1955), 285-309.

[9] H. Turner, "Signed logic programs", In Maurice Bruynooghe, editor, Logic Programming: Proc. of the 1994 Intl. Symposium, pp. 61-75. MIT Press, 1994.

[10] M.A. Khamsi, D. Misane, and V. Kreinovitch, "A new method of proving the existence of answer sets for disjunctive logic programs", In C. Baral and M. Gelfond (Edts.), Proc. ILPS'93 Workshop on Logic Programming with Incomplete Information, 1993.

[11] M.A. Khamsi, and D. Misane, "Disjunctive Signed Logic Programs", Preprint.