Disjunctive Signed Logic Programs

Mohamed A. Khamsi[1] and Driss Misane[2]

[1]*Department of Mathematical Sciences*

University of Texas at El Paso, El Paso, TX 79968.

[2]*Département de Mathématiques, Faculté des Sciences*

Université Mohammed V, Rabat, Morocco.

[1]*email: mohamed@math.utep.edu*

[1]*URL: http://www.drkhamsi.com*

Abstract

In this work, we define signed disjunctive programs and investigate the existence of answer sets for this class of programs. Our main argument is based on an analogue to Tarski-fixed point theorem which we prove for multivalued mappings. This is an original approach compared to known techniques used to prove the existence of answer sets for disjunctive programs.

1 Introduction

In order to introduce a semantics in logic programing, one would in general define an operator for which its fixpoints define the introduced semantics. Therefore, it is important to know whether the given operator has a fixpoint. One will always look for traditional fixpoint theorems and try to see whether the given operator satisfies the assumptions that will insure the existence of fixpoints. Since the beginning of the logic programming theory, the main fixpoint result used is the classical Tarski Theorem. The main assumption in this theorem is monotonicity. Therefore the use of this theorem was extensive in the monotone theory (positive programs). The introduction of the classical negation showed the limitation in the use of Tarski Theorem. Therefore one of the main problems in this direction, is to come up with general fixpoint results to substitute Tarski's theorem. This is the reason why many were trying to prove the existence of the fixpoint of the associated operator directly in reference to a general theorem. Recently other directions were developed: topological, metric [2,5]. It is very exiting to see that the metric approach was very successful. The power behind metric's theorems is the overcome of the monotonicity making the use of classical negation a natural one.

Since the beginning, our research was focused on the study of disjunctive programs. Recall that the operator defining the given semantics is multivalued, i.e. for every x, $T(x)$ is a set which may contains more than one element. To the best of our knowledge, the fixpoint theory is mainly single valued oriented. For example, we do not know of any multivalued fixpoint theorem similar to the classical Tarski theorem. Therefore our approach was to investigate a multivalued extension of Tarski theorem.

In this paper, we give a result that can be seen as a multivalued Tarski theorem. Then we introduce the class of disjunctive signed programs and show how the use of our theorem makes it possible to discuss the existence of the fixpoint of Gelfond-Lifschitz operator(stable semantics). Recall that the notion of a signed logic program was introduced by Gelfond and Lifschitz [4] after a different formulation given initially by Kunen [6]. Indeed Kunen defined the notion of signing on the predicate dependency graph of a finite first-order program in order to prove that two-valued and three-valued completion semantics coincide on the class of strict normal programs. Gelfond and Lifschitz extended this notion to logic programs in order to prove the soundness translation from a high-level language (used for reasoning about actions) to nondisjunctive logic programs. Turner [9] studied signed programs after generalizing the original definition to programs with classical negation. Many positive results were discovered about signed programs among them the restricted monotonicity. Another beautiful application of signed programs was discovered by Turner and Lifschitz [7] for order-consistent normal programs. Recall that a program is called order-consistent if the natural order defined on the predicate dependency graph is well-founded (for more on this see [1]). Fages [1] was the first to prove that such programs have an answer set. His proof is very complicated. Using the notion of splitting, Turner and Lifschitz were able to prove that order-consistent programs are exactly the programs that split into signed subprograms. This result made it possible to give an easy proof to Fages result. The next step was to generalize the notion of signing to disjunctive programs. Turner was the first to give a generalization. The fact that he assumes that one of the subprograms generated by the signing is nondisjunctive is restrictive. In this work we give a generalization of what would be a disjunctive signed program. The signed disjunctive programs according to Turner are called signed semi-disjunctive programs in this paper.

In this work, we prove the existence of consistent answer set for safe signed semi-disjunctive programs via our multivalued Tarski theorem. This result generalizes Turner's one for head consistent programs. An example is given of a safe signed semi-disjunctive program with no head consistent cover. It is our belief that the application of our multivalued Tarski theorem goes beyond the class of signed disjunctive programs. We hope that this theorem will have an impact in disjunctive theory similar to the one the classical Tarski theorem had in the nondisjunctive case.

2 Preliminaries

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

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

where $l_i \in Lit$. Set

$$Head(r) = \{l_1, \cdots, l_n\}\ , \quad Pos(r) = \{l_{n+1}, \cdots, l_m\}$$

and

$$Neg(r) = \{l_{m+1}, \cdots, l_k\}$$

The rule r is said to be disjunctive if $n \geq 2$, i.e. $Head(r)$ has more than one element and nondisjunctive if $n = 1$. An extended (disjunctinve) program Π is a set of (disjunctive) rules.

To define the answer set semantics of extented programs, let us first consider programs without negation by failure *not*.

Let Π be a program (disjunctive or not) for which $Neg(r)$ is empty for every $r \in \Pi$. A subset X of Lit, i.e $X \in 2^{Lit}$, is said to be closed by rules in Π if for every $r \in \Pi$ such that $Pos(r) \subset X$, we have $Head(r) \cap X$ is not empty.

The set $X \in 2^{Lit}$ is an answer set of Π whenever X is closed by rules and

1. if X contains complementary literals, then $X = Lit$.
2. X is minimal, i.e. if $A \subset X$ and A is closed by rules in Π then $A = X$.

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 disjunctive program that may contain *not* . For $X \in 2^{Lit}$, consider the program Π^X defined by

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

Clearly the program Π^X does not contain *not* and $\alpha(\Pi^X)$ is therefore defined. Gelfond and Lifschitz [3] introduced the operator $GL : 2^{Lit} \rightarrow 2^{Lit}$ defined by $GL(X) = \alpha(\Pi^X)$. The set $X \in 2^{Lit}$ is said to be an answer set of Π if $X \in \alpha(\Pi^X)$, i. e. $X \in GL(X)$. If Π is nondisjunctive then X is an answer of Π if and only if $X = GL(X)$. In both cases, we talk about fixed point of GL.

We will say that $X \in 2^{Lit}$ is consistent if it does not contain complementary literals. The program Π is said to be consistent if there exists a consistent answer set.

A special case of consistent programs are head consistent programs. Recall that the head of the program Π is the set

$$Head(\Pi) = \{l;\ \text{there exists a rule } r \in \Pi \text{ such that } l \in Head(r)\ \} = \bigcup_{r \in \Pi} Head(r)$$

Π is said to be head consistent if $Head(\Pi)$ is consistent. Another class of consistent programs is given by what we call safe programs. Before we give the definition of such programs, we need the following definition.

Definition. Let Π be a disjunctive program and $X \in 2^{Lit}$. We will say that X activates a rule $r \in \Pi$ if and only if $Neg(r) \cap X = \emptyset$ and

$$Pos(r) \subset X \implies Head(r) \cap X \neq \emptyset\ .$$

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 $\Pi = \Pi_1 \cup \Pi_2$.

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 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:

$$\Pi_1 \left\{ \begin{array}{l} ab(f,b,X) \leftarrow p(X) \\ p(s) \leftarrow \end{array} \right.$$

$$\Pi_2 \left\{ \begin{array}{l} f(X) \leftarrow b(X),\ not\ ab(f,b,X) \\ b(X) \leftarrow p(X) \\ \neg f(X) \leftarrow p(X) \\ \neg f(X) \leftarrow \neg b(X) \\ b(t) \leftarrow \\ \neg p(t) \leftarrow \end{array} \right.$$

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

The following two basic properties of answer sets are fundamental for this work.

Proposition 1. Let Π be a disjunctive program.

1. Let $X \in 2^{Lit}$ and $A \in \alpha(\Pi^X)$. Then for every $l \in A$, there exists a rule $r \in \Pi$ such that

$$Pos(r) \subset A\ , \quad Neg(r) \cap A = \emptyset \ \text{ and } \ Head(r) \cap A = \{l\}\ .$$

2. If $X \subset Y$, $X, Y \in 2^{Lit}$, then

$$\forall A \in \alpha(\Pi^X)\ \exists B \in \alpha(\Pi^Y) \quad B \subset A$$

The proof of 1 is based on the minimality of A. For 2, it is not hard to see that A is closed by the rules of Π^Y, therefore it contains an answer set of Π^Y. When the program Π is nondisjunctive, we have

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

in another word the operator GL is anti-monotone.

3 A multivalued Tarski theorem

The main ingredient used to prove the existence of fixpoints of operators in nondisjunctive logic programming is the Tarski theorem. It is worth to mention that these operators are single valued. When one tries to copy these techniques in the disjunctive case we are facing the situation of how to prove the existence of fixpoints of multivalued operators. The first approach is to look for a classical theorem that holds and try to use it. It is unfortunate that, to the best of our knowledge, no formulation of Tarski theorem is known in the multivalued case.

Let $(L, \leq)$ be a complete lattice. Define the relation $\prec_r$ (r for restriction) in 2^L 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). The preorder $\prec_r$ was naturally considered in view of proposition 1.
In a similar fashion, we also define the extension preorder $\prec_e$ in 2^L by

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

Although $\prec_r$ and $\prec_e$ are not orders in general, when restricted to the set

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

they become an order. Recall that an antichain is an unordered set, i.e any two elements are not comparable. A natural example of an antichain is given by $\alpha(\Pi^X)$.

We say that a multivalued mapping T from L into 2^L is $\prec_r$-increasing (in logic programming language we will say T is monotone) if

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

and $\prec_r$-decreasing (in logic programming language we will say T is antimonotone) if

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

Similar definitions can be given for $\prec_e$-increasing and $\prec_e$-decreasing.

Note that proposition 1 implies that the operator GL is $\prec_r$-decreasing (antimonotone) but it is fairly easy to come up with an example which shows that GL is not $\prec_e$-decreasing.

Before we state the main result of this section, we need the following definiton.

Definition. Let L be an ordered set and T a multivalued mapping from L into 2^L. We say that the family (x_β) is a decreasing T-orbit if

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

The next result is a multivalued version of the classical Tarski theorem.

Theorem 1 (Tarski multivalued) Let L be a complete lattice and T be an $\prec_r$-increasing multivalued mapping from L into 2^L such that for every $x \in L$

1. $T(x)$ is not empty (i.e. $T(x) \neq \emptyset$)
2. For every decreasing T-orbit (x_β), there exists $x \in L$ 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 $x \in L$ such that $x \in T(x)$.

Proof. Let x_0 be the greatest element of L ($x_0 = 1$). Since $T(x_0)$ is not empty, choose $x_1 \in T(x_0)$. We have $x_1 \leq x_0$. Since T is $\prec_r$-increasing, then $T(x_1) \prec_r T(x_0)$. Therefore there exists $x_2 \in T(x_1)$ such that $x_2 \leq x_1$, and so on...Therefore we can construct a sequence (x_n) such that

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

If there exists n such that $x_{n+1} = x_n$, the process stops and we reach the fixed point. Otherwise let $x_\omega = \inf(x_n)$, since (x_n) is a decreasing T-orbit there exists $x_{\omega+1} \in L$ such that

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

Therefore by a transfinite induction we can construct a familly (x_α), where α is an ordinal, such that

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

Since (x_α) is decreasing, it can not be strictly decreasing (by ZF). In an other word, we can not have $x_\alpha \neq x_\beta$ for every $\alpha \neq \beta$. Therefore there exists α_0 such that $x_{\alpha_0+1} = x_{\alpha_0}$, that is $x_{\alpha_0} \in T(x_{\alpha_0})$. The proof of Theorem 1 is therefore complete.

Remark. The second hypothesis is a technical one needed in the proof. Note that if $T(x)$ is finite for every $x \in L$, then it is obviously satisfied.

4 Signed disjunctive programs

Signed programs were introduced and used to prove the existence of answer sets for nondisjunctive programs. Turner was the first to attempt a general definition for signing in the disjunctive case. We think that his definition does not carry the general case. In the following we present a definition for signed disjunctive program which we believe is more general.

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. $Holds(f, Result(a, s)) \ \leftarrow \ Holds(f, s), \ notNoinertial(f, a, s)$
8. $\neg Holds(f, Result(a, s) \ \leftarrow \ \neg Holds(f, s), \ notNoinertial(f, a, s)$
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. This program is obviously head consistent and therefore it is a safe program.

Turner [9] using the notion of covers of a disjunctive program proved that signed disjunctive programs have a consistent answer set provided they have at least one head-consistent cover.

In what follows we prove the existence of a consistent answer set for a larger class of signed disjunctive programs. This proof is based on a new fixed point theorem that can be seen as a multivalued Tarski theorem. It is, to our knowledge, the first kind of result that may have a potential application in the study of disjunctive programs as Tarski's theorem did for nondisjunctive programs.

Let Π be a signed disjunctive program with Π_s and $\Pi_{\bar{s}}$ be the associated subprograms where Π_s is nondisjunctive. For any $X \in 2^{Lit}$ clearly we have $\Pi^X = \Pi_s^X \cup \Pi_{\bar{s}}^X$ and $\Pi_s^X = \Pi_s^{X \cap \bar{S}}$ and $\Pi_{\bar{s}}^X = \Pi_{\bar{s}}^{X \cap S}$. Let $L = 2^{\bar{S}}$ and define $T : L \to 2^L$ by

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

Lemma 2: T is $\prec_r$-monotone.

Obvious since the operator $U(X) = \alpha(\Pi^X)$ is anti-monotone.

Lemma 3: Let (X_β) be a decreasing T-orbit in L and denote $Z = \bigcap\limits_\beta X_\beta$.Then there exists $Y \subset \bar{S}$ such that

$$\begin{cases} Y \in T(Z) \\ Y \subset Z. \end{cases}$$

Indeed let us note that Z is closed by rules in $\Pi_{\bar{s}}^{\alpha(\Pi_s^Z)}$. Let $r \in \Pi_{\bar{s}}^{\alpha(\Pi_s^Z)}$ and suppose that $Pos(r) \subset Z$. Since $Z \subset X_\beta$, then

$$\alpha(\Pi_s^{X_\beta}) \subset \alpha(\Pi_s^Z) \quad \text{and} \quad \Pi_{\bar{s}}^{\alpha(\Pi_s^Z)} \subset \Pi_{\bar{s}}^{\alpha(\Pi_s^{X_\beta})} \ .$$

On the other hand, $Pos(r) \subset X_{\beta+1}$ since $Z \subset X_{\beta+1}$. But $X_{\beta+1} \in \alpha(\Pi_{\bar{s}}^{\alpha(\Pi_s^{X_\beta})})$ and since $r \in \Pi_2^{\alpha(\Pi_s^{X_\beta})}$ there exists $A_{i_\beta} \in Head(r)$ such that $A_{i_\beta} \in X_{\beta+1}$. Therefore for every β there exists $A_{i_\beta} \in Head(r)$ such that $A_{i_\beta} \in X_{\beta+1}$. Since $Head(r)$ is finite and the familly (X_β) is infinite, there exists $A \in Head(r)$ such that $A \in X_\beta$ for all β.

Therefore Z is closed by rules in $\Pi_{\bar{s}}^{\alpha(\Pi_s^{X_2})}$. This clearly implies the existence of $Y \subset Z$ such that $Y \in \alpha\Big(\Pi_{\bar{s}}^{\alpha(\Pi_s^Z)}\Big)$.

Theorem 2. Let Π be a signed safe semi-disjunctive program. Then Π has a consistent answer set.

Proof: Because of Theorem 1 the operator T has a fixed point, i.e. there exists $X_2 \in 2^{\bar{S}}$ such that $X_2 \in T(X_2) = \alpha(\Pi_{\bar{s}}^{\alpha(\Pi_s^{X_2})})$. Let $X_1 = \alpha(\Pi_s^{X_2})$. Let us prove that $X = X_1 \cup X_2$ is a consistent answer set of Π. Indeed, notice that if $l \in X_1 \cup X_2$,then there exists a rule $r \in \Pi$ activated by X_1 such that $(X_1 \cup X_2) \bigcap head(r) = \{l\}$.
Indeed if $l \in X_1$ because of Proposition 1 there exists a rule r such that $pos(r) \subset X_1$ and $head(r) \cap X_1 = \{l\}$. But $head(r) \subset S$ therefore $head(r) \bigcap (X_1 \cup X_2) = head(r) \cap X_1$. This clearly implies that X_1 activates r. On the other hand, if $l \in X_2$, then there exists a rule r such that $pos(r) \subset X_2$ and

$head(r) \cap X_2 = \{l\}$. But $head(r) \subset \bar{S}$ therefore $head(r) \bigcap (X_1 \cup X_2) = head(r) \cap X_2$. Since $X_2 = \alpha(\Pi_{\bar{s}}^{X_1})$, then $Neg(r) \cap X_1 = \emptyset$, otherwise the rule r will be removed. Clearly X_1 activates r. Let us now show that $X_1 \cup X_2$ is consistent. Assume that there exists a literal l such that l and $\neg l$ are in $X_1 \cup X_2$. Therefore there exists two rules r_1 and r_2 activated by X_1 such that

$$head(r_1) \bigcap (X_1 \cup X_2) = \{l\} \quad \text{and} \quad head(r_2) \bigcap (X_1 \cup X_2) = \{\neg l\}$$

This clearly contradicts the fact that X_1 can not activate two contrary rules. Let us now prove that $X_1 \cup X_2$ is an answer set of Π. First at all, let us show that $X_1 \cup X_2$ is closed by rules of $\Pi^{X_1 \cup X_2}$.
Let r be a rule in $\Pi^{X_1 \cup X_2}$ such that $pos(r) \subset X_1 \cup X_2$. Since $\Pi_s^{X_2}$ and $\Pi_{\bar{s}}^{X_1}$ are disjoints and $\Pi^{X_1 \cup X_2} = \Pi_s^{X_2} \bigcup \Pi_{\bar{s}}^{X_1}$, then $r \in \Pi_{\bar{s}}^{X_1}$ or $r \in \Pi_s^{X_2}$.

(1) If $r \in \Pi_{\bar{s}}^{X_1}$ then $pos(r) \subset \bar{S}$ i.e. $pos(r) \subset X_2$, and since $X_2 \in \alpha(\Pi_{\bar{s}}^{X_1})$ then $head(r) \cap X_2 \neq \emptyset$.Therefore $head(r) \bigcap (X_1 \cup X_2) \neq \emptyset$.

(2) If $r \in \Pi_s^{X_2}$ we do the same thing.

Now let us prove that $X_1 \cup X_2$ is minimal. Indeed, suppose that there exists $Y \subset X_1 \cup X_2$ closed by rules in $\Pi^{X_1 \cup X_2} = \Pi_s^{X_2} \bigcup \Pi_{\bar{s}}^{X_1}$. Denote $Y_1 = Y \cap X_1$, then $Y_1 \subset X_1$ and it is easy to show that $Y_1 = \alpha(\Pi_s^{X_2})$; therefore $Y_1 = X_1$. In the same way we have $Y_2 = Y \cap X_2 \in \alpha(\Pi_{\bar{s}}^{X_1})$, then $Y_2 = X_2$. Therefore $Y = Y_1 \cup Y_2 = X_1 \cup X_2$.

Remark. In fact the hypothesis that Π_1 is a nondisjunctive program is not necessary. Indeed, it is sufficient that only $\alpha(\Pi_1^X)$ is assumed to be a singleton for any $X \in 2^{\bar{S}}$. An example of this is given by

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

A signing for this program is $S = \{a, b\}$. Consider the operator T defined in the proof of Theorem 2. 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.

References

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

[2] M. Fitting: Metric methods, three examples and a theorem. Journal of Logic Programming, 1993.

[3] M. Gelfond and V. Lifschitz: Classical negation in logic programs and disjunctive databases. New generation Computing, 9:365-385, 1991.

[4] M. Gelfond and V. Lifschitz: Representating action and change by logic programs. The Journal of Logic Programming, 17:301-322, 1993.

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

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

[7] V. Lifschitz and H. Turner: Splitting a logic program. In Logic Programming: Proceedings of the Eleventh International Conference, 1994. To appear.

[8] A. Tarski: A lattice-theoretical fixpoint theorem and its applications. Pacific Journal of Mathematics, 5:285-309, 1955.

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