IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661, ISBN: 2278-8727Volume 7, Issue 3 (Nov. - Dec. 2012), PP 41-50
www.iosrjournals.org

Justified Answer Set Programming

[1]Amr I. Hassan, [2]Mona Gharib
[1]*Mathematics and Computer Science Department, Faculty of Science, Port Said University, Port Said, Egypt.*
[2]*Mathematics Department, Faculty of Science, Zagazig University, Zagazig, Egypt.*

Abstract: *In answer set programming, the existence of an answer set for a logic program is not guaranteed. In order to remedy this problem, an incremental answer sets have been introduced. In this paper a concept of a justified answer set is introduced. The idea is to obtain a construct similar to justified extensions of default theories with a semi-monotonicity fixed point operator, and similarly to the concept of incremental answer sets to guarantee existence of an extension. Furthermore, at the level of fixed-points, we establish a one-to-one correspondence between justified answer sets of a logic program and justified extensions of the default theory.*
Keywords: *Default Theory, Fixed-Points, Justified Answer Set, Logic Program, Semi-Monotonicity Property.*

I. Introduction

Answer Set Programming (ASP; [1]) has emerged as an attractive paradigm for declarative problem solving ([11],[14],[15])Originally, it was developed as a declarative branch of logic programming [12], where the semantics of logic programs is given by their answer sets [8]. The answer set semantics is closely related to other non monotonic formalisms, such as Reiter's default logic [16] and Clark's completion [4]. Similar to them, the existence of an answer set for a logic program is not guaranteed. In order to remedy this problem, an incremental answer set approach (ι-answer sets) has been proposed in ([5], [6], [7],[9]) In this paper, we introduce justified answer sets extension of ι-answer sets. The respective concept of justified answer sets is defined by the fixed-points construction.

It leads to that a justified answer set of a logic program is a pair of sets of atoms, and any logic program having at least one justified answer set. The concept of justified answer sets guarantees their existence for every logic program by a property often called semi-monotonicity in the context of default logic. Furthermore, at the level of fixed-points, we establish a one-to-one correspondence between justified answer sets of a logic program and Łukaszewicz (justified) extensions [13] of the default theory. The outline of this paper is as follows. The second section provides some basic concepts. In the third section, we introduce the concept of a justified answer set and elaborate its formal properties. In the fourth section, we characterize the relationship between justified answer sets and ı-answer sets. In the fifth section, we show that there is a one-to-one correspondence between justified answer sets and Łukaszewicz extensions.

II. Background

A (normal) logic program is a finite set of rules of the form

$$a_0 \leftarrow a_1, \dots, a_m, not\ a_{m+1}, \dots, \quad not a_n \qquad (1)$$

where $n \geq m \geq 0$, and each a_i $(0 \leq i \leq n)$ is an atom.

Given a rule r as in (1), we denote the head of r by $head(r) = a_0$ and the body of r by $body(r) = \{a_1, \dots, a_m, not\ a_{m+1}, \dots, not a_n\}$. Furthermore, we let $body^+(r) = \{a_1, \dots, a_m\}$ and $body^-(r) = \{a_{m+1}, \dots, a_n\}$ be the positive and negative body of r, respectively. For a logic program Π, we let $body^+(\Pi) = \cup_{r \in \Pi} body^+(r)$ and $body^-(\Pi) = \cup_{r \in \Pi} body^-(r)$. A literal is either an atom or a negated atom. A program is called basic if $body^-(r) = \emptyset$ for all $r \in \Pi$. A set X of atoms is closed under a basic program Π if, for any $r \in \Pi$, head(r) $\in$ X whenever $ody^+(r) \subseteq X$. The smallest set of atoms which is closed under a basic program is denoted by $Cn(\Pi)$. The redact of a logic program relative to a set X of atoms is $\Pi^X = \{head(r) \leftarrow body^+(r) \mid r \in \Pi\ and\ body^-(r) \cap X = \emptyset\}$. A set X of atoms is an answer set of if $X = Cn(\Pi^X)$. For a program Π, we let $Cn^+(\Pi) = Cn(\Pi^{\emptyset})$. Note that $\Pi^{\emptyset} = \{head(r) \leftarrow body^+(r) \mid r \in \Pi\}$. Alternative inductive characterizations for the operator Cn can be obtained by appeal to immediate consequence operators [12]. For a logic program and a set X of atoms, the operator $T_{\Pi}(X) = \{head(r) \mid r \in \Pi,\ body^+(r) \subseteq X\ and\ body^-(r) \cap X = \emptyset\}$. Iterated applications of T_{Π} are written as T_{Π}^{j} for $j \geq 0$, where $T_{\Pi}^0(X) = X$ and $T_{\Pi}^i(X) = T_{\Pi}\left(T_{\Pi}^{i-1}(X)\right) for\ i \geq 1$. It is well-known that $Cn(\Pi) = \cup_{i \geq 0} T_{\Pi}^i(\emptyset)$, for any basic program Π. Also, for any answer set X of program Π, it holds $X = \cup_{i \geq 0} T_{\Pi^X}^i(\emptyset)$. For a program Π and a set X of atoms, the generating rules of X for Π are

$R_{\Pi}(X) = \{r \in \Pi \mid body^+(r) \subseteq X\ and\ body^-(r) \cap X = \emptyset\}$. In fact, one can show that X is an answer set of Π iff $X = Cn^+(R_{\Pi}(X)) = Cn(\Pi^X) = X$.

III. Justified Answer Sets

In answer set programming, the existence of an answer set is not guaranteed. Answer sets are defined via a reduction of logic programs to basic program. It means that we cannot determine which rules make blocking with each other. So we reduct the logic program according to a maximal subset of rules. This reduct remove complete rules according to pair of sets of atoms. Now, we define the reduct of the logic program according to pair of sets of atoms as follows.

Definition 3 .1 Let be a logic program and let X and Y be sets of atoms.
The reduct of Π relative to (X, Y) is

$$\Pi^{(X,Y)} = \{r \mid r \in \Pi, (X \cup \{head(r)\}) \cap (Y \cup body^{-}(r)) = \emptyset\}.$$

For illustration, consider the following logic program Π_1:

$r_1: a \leftarrow a,$
$r_2: b \leftarrow not a,$ (2)
$r_3: c \leftarrow.$

We reduct this program according two sets as follows:

(X, Y)	$\Pi^{(X,Y)}$
$(\{\}, \{\})$	$r_1: a \leftarrow a,$ $r_2: b \leftarrow not a$ $r_3: c \leftarrow.$
$(\{a\}, \{\})$	$r_1: a \leftarrow a,$ $r_3: c \leftarrow.$
$(\{b\}, \{\})$	$r_1: a \leftarrow a,$ $r_2: b \leftarrow not a$ $r_3: c \leftarrow.$
$(\{\}, \{a\})$	$r_2: b \leftarrow not a$ $r_3: c \leftarrow.$
$(\{\}, \{b\})$	$r_1: a \leftarrow a,$ $r_3: c \leftarrow.$
$(\{a\}, \{b\})$	$r_1: a \leftarrow a,$ $r_3: c \leftarrow.$
$(\{a, b\}, \{\})$	$r_1: a \leftarrow a,$ $r_3: c \leftarrow.$
$(\{b\}, \{a\})$	$r_2: b \leftarrow not a$ $r_3: c \leftarrow.$
$(\{\}, \{a, b\})$	$r_3: c \leftarrow.$
$(\{c\}, \{\})$	$r_1: a \leftarrow a,$ $r_2: b \leftarrow not a$ $r_3: c \leftarrow.$
$(\{a, c\}, \{b\})$	$r_1: a \leftarrow a,$ $r_3: c \leftarrow.$
$(\{b, c\}, \{a\})$	$r_2: b \leftarrow not a$ $r_3: c \leftarrow.$

According to a set X of atoms, the negative body of each $r \in \Pi$ such that $body^{+}(r) \subseteq X$ and $body^{-}(r) \cap X = \emptyset$ can be defined as follows.

Definition 3.2 Let Π be a logic program and let X be a set of atoms.
We define the function $S_{\Pi}(X)$ from a set of atoms to a set of atoms as
$S_{\Pi}(X) = \{ q \mid r \in \Pi, body^{+}(r) \subseteq X, q \in body^{-}(r) \text{ and } body^{-}(r) \cap X = \emptyset\}.$

We introduce justified answer sets as pair of sets of atoms this pair of sets are fixpoint of an operator. So, first we define a consequence operator that induces a pair of sets of atoms as follows.

Definition 3.3 Let Π be a logic program and let X be set of atoms.
The consequence operator $\tau_{\Pi}(X)$ is $\tau_{\Pi}(X) = (T_{\Pi}(X), S_{\Pi}(X)).$

We observe that operator τ_{Π} induces a pair (X', Y') of sets of atoms. For comparing two pairs of sets of atoms (X, Y) and (X', Y'), we define $(X', Y') \sqsubseteq (X, Y)$ if $X' \subseteq X$ and $Y' \subseteq Y$. Then, operator τ_Π is monotonic in the following sense.

Theorem 3.1 Let be a logic program and let X and X' be two sets of atoms.
If $X' \subseteq X$ then $\tau_{\Pi}(X') \sqsubseteq \tau_{\Pi}$(X)

Since operator τ_{Π} is monotonic, it has a least fixedpoints [17]. We denote the least fixed-points of τ_{Π} by $Cn_j(\Pi)$**.** Observe that the second argument of $Cn_j(\Pi)$ is completely determined by the first one.

Theorem 3.2 Let Π be a logic program and let X be a set of atoms.
Then, $Cn_j(\Pi) = (X, S_{\Pi}(X))$ for $X = Cn^+(\Pi)$.

For computing $Cn_j(\Pi)$**.** we may start with the empty set and iterate τ_{Π} until its least fixed-points is reached. Iterated applications of τ_{Π} are written as $\tau_{\Pi}^{j}\ for\ j \geq 0$, where $\tau_{\Pi}^{0}(X) = (X, Y)$ and $\tau_{\Pi}^{i+1}(X) = \tau_{\Pi}\left(\tau_{\Pi}^{i}(X)\right) for\ i \geq 0$.
For two pairs of sets of atoms (X, Y) and (X', Y'), we define
(X, Y) $\sqcup$ (X', Y'), as $(X \cup X', Y \cup Y')$. Thus, according to the Knaster-Tarski theorem [17], we conclude that.

Corollary 3.3 For any logic program Π,

$$Cn_j(\Pi) = \coprod_{i \geq 0} \tau_{\Pi}^{i}(\emptyset).$$

Consequently, we introduce the relationship between the reduct of a logic program according to a pair of sets of atoms and the consequence operator as following.

Theorem 3.4 Let Π be a logic program and let X and Y be sets of atoms.
Then, we have that $X = Cn^+(\Pi^{(X,Y)})$ and $Y = S_{\Pi^{(X,Y)}}(X)$ iff (X, y) is the least fixpoint of $\tau_{\Pi^{(X,Y)}}$.

Since operator $\tau_{\Pi}(X)$ has a least fixpoint, then by Definition 3.3 and [12]. We determine the least fixpoint by $T_{\Pi}(X)$and $S_{\Pi}(X)$. So we conclude that.

Corollary 3.5 Let Π be a logic program and (X, Y) be a pair of sets of atoms.
If (X, Y) is a justified answer set of Π, then

$$(X, Y) = \coprod_{i \geq 0} \tau_{\Pi^{(X,Y)}}^{i}(\emptyset).$$

In view of Corollary 3.3 and Theorem 3.4, we define justified answer sets of a logic program as follows.

Definition 3.4 Let Π be a logic program and let X and Y be sets of atoms.
Then, X is a justified answer set of Π with respect to Y iff $X = Cn^+(\Pi^{(X,Y)})$ and $Y = S_{\Pi^{(X,Y)}}(X)$ such that
1. $body^+\left(\Pi^{(X,Y)}\right) \subseteq Cn^+(\Pi^{(X,Y)})$ and
2. $body^+\left(\Pi^{(X,Y)}\right) \cap Cn^+\left(\Pi^{(X,Y)}\right) = \emptyset$.
This shows that a justified answer set of a logic program is a pair of sets of atoms. Definition 3.4 characterizes justified answer sets (X, Y) of Π in terms of the rules that apply with respect to (X, Y). The set $\Pi^{(X,Y)}$ of such rule is maximal among all subsets of that satisfy conditions (1) and (2). Condition (1) guarantees that the positive bodies of rules in $\Pi^{(X,Y)}$are justified, while condition (2) makes sure that the rules in $\Pi^{(X,Y)}$ do not block one another. That is, $(X, Y) = Cn_j(\Pi^{(X,Y)}) = (X, S_{\Pi^{(X,Y)}}(X))$ where $X = Cn^+\left(\Pi^{(X,Y)}\right)$.

For illustration, reconsider program Π_1 in (2), consisting of rules:

$$r_1\colon a \leftarrow a,$$
$$r_2\colon b \leftarrow nota$$
$$r_3\colon c \leftarrow.$$

This Program has $(\{b,c\},\{a\})$ as its unique justified answer set, as can be verified by means of the following table:

(X,Y)	$\Pi^{(X,Y)}$	$Cn^{+}(\Pi^{(X,Y)})$	$S_{\Pi^{(X,Y)}}(X)$
$(\{\ \},\{\ \})$	$r_1: a \leftarrow a,$ $r_2: b \leftarrow not a$ $r_3: c \leftarrow.$	$\{b,c\}$	$\{a\}$
$(\{a\},\{\})$	$r_1: a \leftarrow a,$ $r_3: c \leftarrow.$	$\{c\}$	$\{\ \}$
$(\{b\},\{\})$	$r_1: a \leftarrow a,$ $r_2: b \leftarrow not a$ $r_3: c \leftarrow.$	$\{b,c\}$	$\{a\}$
$(\{\},\{a\})$	$r_2: b \leftarrow not a$ $r_3: c \leftarrow.$	$\{b,c\}$	$\{a\}$
$(\{\},\{b\})$	$r_1: a \leftarrow a,$ $r_3: c \leftarrow.$	$\{c\}$	$\{\ \}$
$(\{a\},\{b\})$	$r_1: a \leftarrow a,$ $r_3: c \leftarrow.$	$\{c\}$	$\{\ \}$
$(\{a,b\},\{\})$	$r_1: a \leftarrow a,$ $r_3: c \leftarrow.$	$\{c\}$	$\{\ \}$
$(\{b\},\{a\})$	$r_2: b \leftarrow not a$ $r_3: c \leftarrow.$	$\{b\}$	$\{a\}$
$(\{\},\{a,b\})$	$r_3: c \leftarrow.$	$\{c\}$	$\{\ \}$
$(\{c\},\{\})$	$r_1: a \leftarrow a,$ $r_2: b \leftarrow not a$ $r_3: c \leftarrow.$	$\{b,c\}$	$\{a\}$
$(\{a,c\},\{b\})$	$r_1: a \leftarrow a,$ $r_3: c \leftarrow.$	$\{c\}$	$\{\ \}$
$(\{b,c\},\{a\})$	$r_2: b \leftarrow not a$ $r_3: c \leftarrow.$	$\{b,c\}$	$\{a\}$

In the following, we provide some properties of logic programs and their justified answer sets. A fundamental result asserts that justified answer sets of a logic programs are extended when additional rules are introduced. This property, often called semi-monotonicity in the context of default logic, can be stated as follows.

Theorem 3.6 Let Π and Π' be two logic programs such that $\Pi' \subseteq \Pi$. Let (X,Y) and (X',Y') be two pairs of sets of atoms.

If $X'$is a justified answer set of Π' with respect to Y', then there exists a justified answer set X of Π with respect to Y such that $X' \subseteq$ X and $Y' \subseteq$ Y.

For illustration, reconsider Program Π_1 in (2) and program Π'=$\{r_3\} \subset \Pi_1$. The program Π' has a justified answer set $(X',Y') = (\{c\},\emptyset)$. We have seen that Π_1 has unique justified answer set $(\mathrm{X},\mathrm{Y}) = (\{\mathrm{b},\mathrm{c}\},\{\mathrm{a}\})$. Then, we conclude that $X' \subseteq X$ and $Y' \subseteq Y$.

As a consequence of Theorem 3.6, we obtain the following result.

Corollary 3.7 Any logic program has a justified answer set.
Now, we proceed to give some results characterizing the relationship among standard answer sets and justified answer sets.

Theorem 3.8 Let Π be a logic program and let X and Y be two sets of atoms.

If X is an answer set of Π, then X is a justified answer set of Π with respect to Y .
In other words, the set of justified answer sets of a logic program Π forms a superset of its answer sets.
For illustration, consider the following logic program Π_3:

$$r_1: a \leftarrow,$$
$$r_2: b \leftarrow not a$$

This program has two justified answer sets: $(\{a\},\emptyset)$ and $(\{b\},\{a\})$.However, $(\{a\})$ is the only classical answer set Π_3.

IV. Relationship to ι-Answer Sets

In this section, we characterize the relationship between justified answer sets and ι-answer sets. In the following, we introduce a justified answer set without fixed-points. The characterization is based on the notion of generating rules. This leads us to the concept of generating rules as follows.

Definition 4.1 Let Π be a logic program and let X and Y be sets of atoms.
We define the set of generating rules of Π with respect to (X, Y) as
$R_{\Pi}(X) = \{ r \in \Pi \mid body^{+}(r) \subseteq X \; and \; (X \cup \{head(r)\}) \cap (Y \cup body^{-}(r)) = \emptyset\}$. From this definition we can derive the following result.

Theorem 4.1 Let Π be a logic program and let X and Y be sets of atoms.
Then, (X, Y) is a justified answer set of Π iff $X = Cn^{+}(R_{\Pi}(X, Y))$ and $Y = body^{-}(R_{\Pi}(X, Y))$.

This result show that the concept of generating rules yields an alternative characterization of justified answer sets. From Definition 4.1 and Theorem 4.1, we obtain the following result.

Theorem 4.2 Let Π be a logic program and let X and Y be sets of atoms.
Then, (X, Y) is a justified answer set of Π iff $X = Cn^{+}(\Pi')$ and $Y = body^{-}(\Pi')$ for some $\Pi' \subseteq \Pi$such that for each $r \in \Pi$

1. $If\ r \in \Pi', then\ body^{+}(r) \subseteq X \; and \; (X \cup \{head(r)\}) \cap (Y \cup body^{-}(r)) = \emptyset\}$.
2. $If\ r \notin \Pi', then\ body^{+}(r) \nsubseteq X \; and \; (X \cup \{head(r)\}) \cap (Y \cup body^{-}(r)) \neq \emptyset\}$.

Theorem 4.2 characterizes a justified answer set (X, Y) of Π in terms of the rules that apply conditions (1) and (2). Thus, the set of rules Π' is maximal among all subsets of Π that satisfy conditions (1) and (2). This leads us to the following result.

Theorem 4 .3 Let Π be a logic program and let X and Y betwo sets of atoms.
Then, X is a justified answer set wrt Y of Π $X = Cn^{+}(\Pi')$ and $Y = body^{-}(\Pi')$ for some ⊆ maximal $\Pi' \subseteq \Pi$such such that

1. $body^{+}(\Pi') \subseteq Cn^{+}(\Pi')$
2. . $body^{-}(\Pi') \cap Cn^{+}(\Pi') = \emptyset$.

From Theorem 4.3, we have that X is a justified answer set wrt Y of Π iff X is an ι-answer set of Π such that $X \cap Y = \emptyset$. So, From Theorem 4.3 and Definition 1 in [7], we conclude that there is a one-to-one corresponding between justified answer sets and ι-answer
sets of logic programs.

V. Relationship to Default Logic

Łukaszewicz [13] modified default logic in order to guarantee the existence of extensions and semimonotonicity for general default theories. The correspondence between Reiter's default logic [16] and logic programming has been exhaustively studied ([2], [3], [8]) this section discusses the relation between the variant of default logic proposed by Łukaszewicz and justified answer sets of logic programs. The definition of justified extensions in [13] is based on fixed-points.

Definition 5.1 [13]Let $\Delta = (D, W)$ be a default theory. For any pair of sets of formulas (S, T), let $\Gamma_{\Delta}(S, T)$ be the pair of smallest sets of formulas (S', T') such that

1. $W \subseteq S'$,
2. 2. $S' = Cn_{\models}(S')$,
3. $For\ any\ \delta = \frac{\alpha:\beta}{\gamma} \in D, if\ \alpha \in S'\ and\ \neg\eta \notin Cn_{\models}(S \cup \{\gamma\})\ for\ all\ \eta \in T \cup \{\beta\}, then\ \gamma \in S'\ and\ \beta \in T'$.

A set E of formulas is a justified extension of Δ with respect to a set J of formulas iff $\Gamma_{\Delta}(E, J) = (E, J)$.

Where $Cn_{\models}(.)$ denotes the deductive closure in the sense of propositional logic. From [10], a logic program Π can be transformed into a default theory by turning each rule r of form in Π into the default rule

$$p_0 \leftarrow p_1, \dots, p_m, not\ p_{m+1}, \dots, \quad not p_n$$

$$\delta_r = \frac{p_1 \wedge \dots \wedge p_m \wedge : \neg p_{m+1} \wedge \dots \wedge \neg p_n}{p_0}$$

We denote the default theory corresponding to Π by $\Delta_\Pi = (\{\delta_r \mid r \in \Pi\}, \emptyset)$. Here, we only consider default theories such that $W = \emptyset$.

Now, at the level of fixed-points, we introduce the relationship between justified answer sets of a logic program Π and the justified extension of the corresponding default theory Δ_Π as follows:

Theorem 5.1 Let Π be a logic program.
1. $If\ (X, Y)\ is\ a\ justified\ answer\ set\ of\ \Pi, then$
$$(Cn_{\models}(X, Y) is\ a\ justified\ extension\ of\ \Delta_\Pi.$$
2. $Every\ justified\ extension\ of\ \Delta_\Pi\ has\ the\ form$
$$(Cn_{\models}(X, Y)\ for\ exactly\ one\ justified\ answer\ set (X, Y)\ of\ \Pi.$$

From Theorem 5.1, we have that there is a one-to-one correspondence between justified answer sets and Łukaszewicz extensions.

For illustration, reconsider Program Π_1 in (2). We have seen that Π_1 has $(\{b, c\}, \{a\})$ as its unique justified answer set, also its corresponding default theory $\Delta_{\Pi_1} = (\left\{\frac{b:\neg c}{a}, \frac{c:\neg a}{c}\right\}, \emptyset)$ has $Cn_{\models}(\{b, c\}, \{a\})$ as its unique justified extension.

VI. Conclusion

In this work, we have elaborated upon the concept of justified answer sets, which is an extension of ι-answer sets [7]. The justified answer set defined by the fixed-points construction having a property, often called semi-monotonicity in the context of default logic. Based on the concept of generating rules, we have shown a one-to-one correspondence between justified answer sets and ι-answer sets of logic programs. The justified answer sets of a logic program amount to the Łukaszewicz (justified) extensions [13] of the default theory corresponding to the program. Similar to justified extensions of default theories but different from (standard) answer sets, every logic program has at least one justified answer set.

VII. Proofs

Proof 3.1 Let be a logic program and let X and X' be two sets of atoms.
such that $X' \subseteq X$

We have to prove that $\tau_{\boldsymbol{\Pi}}(X') \sqsubseteq \tau_{\boldsymbol{\Pi}}(X)$. By Definition 3.3, we have that $\tau_{\boldsymbol{\Pi}}(X) = (T_{\boldsymbol{\Pi}}(X), S_{\boldsymbol{\Pi}}(X))$. and by Definition 3.2, we have that

$S_{\boldsymbol{\Pi}}(X) = \{ q \mid r \in \Pi, body^+(r) \subseteq X, q \in body^-(r) \text{ and } body^-(r) \cap X = \emptyset\}$.

Since

$T_{\boldsymbol{\Pi}}(X) = \{ head(r) \mid r \in \Pi, body^+(r) \subseteq X \text{ and } body^-(r) \cap X = \emptyset\}$.

Then, we have that

$\tau_\Pi(X) = (\{ head(r) \mid r \in \Pi, body^+(r) \subseteq X \text{ and } body^-(r) \cap X = \emptyset\}, =$
$\{ q \mid r \in \Pi, body^+(r) \subseteq X, q \in body^-(r) \text{ and } body^-(r) \cap X = \emptyset\})$

And

$\tau_\Pi(X') = (\{ head(r) \mid r \in \Pi, body^+(r) \subseteq X' \text{ and } body^-(r) \cap X' = \emptyset\}, =$
$\{ q \mid r \in \Pi, body^+(r) \subseteq X', q \in body^-(r) \text{ and } body^-(r) \cap X' = \emptyset\})$.

Since $X' \subseteq X$ then, we have that

$$\{ head(r) \mid r \in \Pi, body^+(r) \subseteq X' \text{ and } body^-(r) \cap X' = \emptyset\}$$
$$\subseteq \{ head(r) \mid r \in \Pi, body^+(r) \subseteq X \text{ and } body^-(r) \cap X = \emptyset\}$$

And

$$\{ q \mid r \in \Pi, body^+(r) \subseteq X', q \in body^-(r) \text{ and } body^-(r) \cap X' = \emptyset\}$$
$$\subseteq \{ q \mid r \in \Pi, body^+(r) \subseteq X, q \in body^-(r) \text{ and } body^-(r) \cap X = \emptyset\}$$

It is follows that $T_\Pi(X') \subseteq T_\Pi(X)$ and $S_\Pi(X') \subseteq S_\Pi(X)$.

Thus, $\tau_\Pi(X') \sqsubseteq \tau_\Pi(X)$.

Proof 3.2 Let Π be a logic program and let X be set of atoms such that $X = Cn^{+}(\Pi)$, then $X = Cn(\Pi^{\emptyset})$. Since $T_{\Pi}(X) = \{ \text{head}(r) \mid r \in \Pi,\ body^{+}(r) \subseteq X \text{ and } body^{-}(r) \cap X = \emptyset\}$, then we have that $X = T_{\Pi}(X)$ so X is the least fixed point of T_{Π}.
By Definition 3.3, we have $\tau_{\Pi}(X) = (T_{\Pi}(X), S_{\Pi}(X))$ and by Theorem 3.1, operator τ_{Π} is monotonic, it has a least fixed point and the second argument of $Cn_j(\Pi)$ determined by the first one. Then, we have that $\left(X, S_{\Pi}(X)\right)$ is the least fixed point of τ_{Π}. Thus $\tau_{\Pi}(X) = \left(X, S_{\Pi}(X)\right))$ and we get that $Cn_j(\Pi) = \left(X, S_{\Pi}(X)\right)$

Proof 3.4 Let Π be a logic program and let X and Y be two sets of atoms. Let $X = Cn^{+}(\Pi^{(X,Y)})$ and $Y = S_{\Pi(X,Y)}(X)$ then, we show that (X, Y) is the least fixed point of $\tau_{\Pi(X,Y)}$. Since $X = Cn^{+}(\Pi^{(X,Y)})$, then X is the least fixed point of $T_{\Pi(X,Y)}$ and we have that $X = \bigcup_{i\geq 0} T^{i}_{\Pi\,(X,Y)}(\emptyset)$. By Definition 3.3, we have that $\tau_{\Pi(X,Y)}(X) = (T_{\Pi(X,Y)}(X), S_{\Pi(X,Y)}(X))$. So
$\left(X, S_{\Pi(X,Y)}(X)\right) = \coprod_{i\geq 0} \tau^{i}_{\Pi}(\emptyset)$. Since $Y = S_{\Pi(X,Y)}(X)$ then, we have that $(X, Y) = \coprod_{i\geq 0} \tau^{i}_{\Pi}(\emptyset)$

<=: Let (X, Y) is the least fixed point of $\tau_{\Pi(X,Y)}$ then, we have to show that $X = Cn^{+}(\Pi^{(X,Y)})$ and $Y = S_{\Pi(X,Y)}(X)$. Since (X, Y) is the least fixed point of
$\tau_{\Pi(X,Y)}$ then $(X, Y) = \coprod_{i\geq 0} \tau^{i}_{\Pi}(\emptyset)$. From Corollary 3.3, we have that $Cn_{j}(\Pi) = \coprod_{i\geq 0} \tau^{i}_{\Pi}(\emptyset)$ and from Theorem3.2, $Cn_{j}\left(\Pi^{(X,Y)}\right) = (X, S_{\Pi(X,Y)}(X))$ for $X = Cn^{+}(\Pi^{(X,Y)})$. So, we have that $(X, Y) = (Cn^{+}\left(\Pi^{(X,Y)}\right), S_{\Pi(X,Y)}(X))$. Its mean that $X = Cn^{+}(\Pi^{(X,Y)})$ and $Y = S_{\Pi(X,Y)}(X)$.

Proof 3.6 Let Π and Π' be two logic programs such that $\Pi' \subseteq \Pi$. Let X' be a justified answer set of Π' with respect to Y'. By Definition 3.4, $X' = Cn^{+}(\Pi'^{(X',Y')})$ and
$Y' = S_{\Pi'(X',Y')}(X')$. such that

1. $body^{+}(\Pi'^{(X',Y')}) \subseteq Cn^{+}(\Pi'^{(X',Y')})$ and
2. $body^{-}(\Pi'^{(X',Y')}) \cap Cn^{+}(\Pi'^{(X',Y')}) = \emptyset$.

We have to show that there exists a justified answer set X of Π with respect to Y such that $X' \subseteq X$ and $Y' \subseteq Y$.
Since, $\Pi'^{(X',Y')} \subseteq \Pi' \subseteq \Pi$ is a subset of Π that satisfies (1)and (2). As a consequence, there exists a pair of sets of atoms (X, Y) such that $\Pi'^{(X',Y')} \subseteq \Pi^{(X,Y)}$ and (1) and (2) hold for $\Pi^{(X,Y)}$. For such $\Pi^{(X,Y)}$, let $X = Cn^{+}(\Pi^{(X,Y)})$ and $Y = S_{\Pi(X,Y)}(X)$. Applying Definition 3.4, X is a justified answer set of Π with respect to . By the (monotonicity) of Cn^{+}, we also have
, $X' = Cn^{+}(\Pi'^{(X',Y')}) \subseteq Cn^{+}\left(\Pi^{(X,Y)}\right) = X$ and $Y' = S_{\Pi'(X',Y')}(X') \subseteq S_{\Pi(X,Y)}(X) = Y$.

Proof 3.8 Let Π be a logic program and let X be a set of atoms such that X is an answer set of Π. We define
$Y = \{q \mid r \in \Pi,\ body^{+}(r) \subseteq X, body^{-}(r) \cap X = \emptyset \text{ and } q \in body^{-}(r)\}$.
We have to prove that X is a justified answer set of Π with respect to Y.
Consider $\tau_{\Pi\,(X,Y)}(U) = (\tau^{+}_{\Pi\,(X,Y)}(U), \tau^{-}_{\Pi\,(X,Y)}(U))$, where

$$\tau^{+}_{\Pi\,(X,Y)}(U) = \{ \text{head}(r) \mid r \in \Pi, body^{+}(r) \subseteq X \textit{ and } (U \cup \{\text{head}(r)\}) \cap (V \cup body^{-}(r)) = \emptyset\}$$

$$\tau^{-}_{\Pi\,(X,Y)}(U)) = \{q \mid r \in \Pi,\ body^{+}(r) \subseteq X, q \in body^{-}(r), q \in body^{-}(r) \textit{ and } (U \cup \{\text{head}(r)\}) \cap (V \cup body^{-}(r)) = \emptyset\}$$

In view of Corollary 3.5, we have that $(X, Y) = \coprod_{i\geq 0} \tau^{i}_{\Pi\,(X,Y)}(\emptyset)$
So, we prove that

a. $X = \bigcup_{i\geq 0} \tau^{+i}_{\Pi\,(X,Y)}(\emptyset) \textit{ and}$
b. $Y = \bigcup_{i\geq 0} \tau^{-i}_{\Pi\,(X,Y)}(\emptyset)$.

We need lemma leading up to our proof.

Lemma 3 .1 If $head(r) \in X$, then the following conditions are equivalent:
$(i) body^{-}(r) \cap X = \emptyset$.
$(ii) (X \cup \{\text{head}(r)\}) \cap (Y \cup body^{-}(r)) = \emptyset$.

Proof Since for any rule such that $body^{+}(r) \subseteq X$ and $body^{-}(r) \cap X = \emptyset$. We have that $head(r) \in X$ and

$body^{-}(r) \in Y$. From here, we get that $X \cap Y = \emptyset, X \cup head(r) = X$ and $Y \cup body^{-}(r) = Y$, So (X ∪{head(r)}) ∩ (Y∪ $body^{-}(r)$) $= \emptyset$**.** Hence, proof of Lemma 3.1 is complete.

Now we return to our proof.
Proof a
Since, X is an answer set of Π, then we have $X = Cn(\Pi^{X}) = \bigcup_{i\geq 0} T^{i}_{\Pi^{X}}(\emptyset)$, where $T^{0}_{\Pi^{X}}(\emptyset) = (\emptyset)$ and $T^{i+1}_{\Pi^{X}}(\emptyset) = T_{\Pi^{X}}(T^{i}_{\Pi^{X}}(\emptyset))$ for all $i \geq 0$. So, by induction on i and by Lemma 3.1 we get that $T^{i}_{\Pi^{X}}(\emptyset) = \tau^{+i}_{\Pi(X,Y)}(\emptyset)$. Hence
$X = \bigcup_{i\geq 0} \tau^{+i}_{\Pi(X,Y)}(\emptyset)$.

Now, we have to prove (b).
By induction on i , we have for all$i \geq 0$, $\tau^{-i}_{\Pi(X,Y)}(\emptyset) \subseteq Y$. This implies $\bigcup_{i\geq 0} \tau^{-i}_{\Pi(X,Y)}(\emptyset) \subseteq Y$. Assume that $\in Y$.
Thus there is a rule $r \in \Pi$ such that $body^{+}(r) \subseteq$ X and
$body^{-}(r) \cap$ X = ø. By (a), $body^{+}(r) \subseteq$ X implies

$$body^{+}(r) \subseteq \tau^{+i}_{\Pi(X,Y)}(\emptyset) \ for\ some\ i \geq 0 \qquad (3)$$

Since, $head(r) \in T^{+i}_{\Pi^{X}}(\emptyset) \subseteq X$. Then, by Lemma 3.1 $body^{-}(r) \cap$ X = ø implies that

$$X \cup\{head(r)\}) \cap (Y\cup body^{-}(r)) = \emptyset \qquad (4)$$

From (3) and (4), we immediately obtain
q $\in \tau^{-i}_{\Pi(X,Y)}(\emptyset) \subseteq \bigcup_{i\geq 0} \tau_{\Pi(X,Y)}(\emptyset)$. Thus, we have $Y \subseteq \bigcup_{i\geq 0} \tau^{-i}_{\Pi(X,Y)}(\emptyset)$.

We need the following lemmas before proving Theorem 4.1

Lemma 4.1 Let Π be a logic program and let X and Y be sets of atoms.
Then, R_{Π}(X, Y) ⊆ $\Pi^{(X,Y)}$.

Proof Let a rule $r \in \Pi$ such that $r \in R_{\Pi}$(X, Y). Then,
$body^{+}(r) \subseteq$ X and (X ∪{head(r)}) ∩ (Y∪ $body^{-}(r)$) = ø. It implies that $r \in \Pi^{(X,Y)}$. Thus we have R_{Π}(X, Y) ⊆ $\Pi^{(X,Y)}$.

Lemma 4 .2 Let X and Y be sets of atoms. If $Cn^{+}(R_{\Pi}(X, Y)) \subseteq$ X, then $Cn^{+}(\Pi^{(X,Y)}) = Cn^{+}(R_{\Pi}(X, Y))$.

Proof Let $Cn^{+}(R_{\Pi}(X, Y)) \subseteq$ X**.**
By R_{Π}(X, Y) ⊆ $\Pi^{(X,Y)}$, we have $Cn^{+}(R_{\Pi}(X, Y)) \subseteq Cn^{+}(\Pi^{(X,Y)})$.

We first show that $Cn^{+}(R_{\Pi}(X, Y))$ is closed under $\Pi^{(X,Y)}$. Since, $Cn^{+}(\Pi^{(X,Y)})$ is the smallest set being closed under $\Pi^{(X,Y)}$ then, we have that $Cn^{+}(\Pi^{(X,Y)}) \subseteq Cn^{+}(R_{\Pi}(X, Y))$ For this, we have to prove that for each $r \in \Pi^{(X,Y)}$ such that $body^{+}(r) \subseteq Cn^{+}(R_{\Pi}(X, Y))$, we have $head(r) \in Cn^{+}(R_{\Pi}(X, Y))$. Let $r \in \Pi^{(X,Y)}$ such that $body^{+}(r) \subseteq Cn^{+}(R_{\Pi}(X, Y))$. Then, we have either
r $\in R_{\Pi}$(X, Y) or r $\notin R_{\Pi}$(X, Y). If r $\in R_{\Pi}$(X, Y) then from Definition.5, we obtain two cases
$Case$ 1: $body^{+}(r) \not\subseteq X$ or
$Case$ 2: (X ∪{head(r)}) ∩ (Y∪ $body^{-}(r)$) ≠ ø.
In case 1, we have $body^{+}(r) \not\subseteq Cn^{+}(R_{\Pi}(X, Y))$, which is a contradiction. In case 2, we have r $\notin \Pi^{(X,Y)}$ but this is a contradiction to $r \in \Pi^{(X,Y)}$. Therefore, let r $\in R_{\Pi}$(X, Y). Then $head(r) \in Cn^{+}(R_{\Pi}(X, Y))$ since $Cn^{+}(R_{\Pi}(X, Y))$ is closed under (R_{Π}(X, Y). Hence, $Cn^{+}(R_{\Pi}(X, Y))$ is closed under $\Pi^{(X,Y)}$.
Thus, we have proven that $X = Cn^{+}(\Pi^{(X,Y)}) = Cn^{+}(R_{\Pi}(X, Y))$.

Now, we are ready to prove Theorem 4.1.
Proof 4.1 Let X is a justified answer set of Π with respect to . Then $X = Cn^{+}(\Pi^{(X,Y)})$ and $Y = S_{\Pi(X,Y)}$(X). We have to show
$(a) X = Cn^{+}(R_{\Pi}(X, Y))$ and
$(b)\ Y = body^{-}$(R_{\Pi}(X, Y)).

Show (a): =>: From Lemma 4.1, we have $R_{\Pi}(X,Y) \subseteq \Pi^{(X,Y)}$. By (monotonicity), $Cn^{+}(\Pi^{(X,Y)}) \subseteq Cn^{+}(R_{\Pi}(X,Y))$. Then, $Cn^{+}(R_{\Pi}(X,Y)) \subseteq X$, so that by applying Lemma 4.2, we conclude that $X = Cn^{+}(\Pi^{(X,Y)}) = Cn^{+}(R_{\Pi}(X,Y))$.
<=: Follows directly by using Lemma4.2.

Show (b): From Definition3.2, we have

$$S_{\Pi^{(X,Y)}}(X) = \{ q \mid r \in \Pi^{(X,Y)}, body^{+}(r) \subseteq X, q \in body^{-}(r)\}$$

$$= \{ q \mid r \in \Pi^{(X,Y)}, body^{+}(r) \ X, (X \cup \{head(r)\}) \cap (Y \cup body^{-}(r)) = \emptyset\ q \in body^{-}(r)\}$$

$$= \{ q \mid r \in R_{\Pi}(X,Y), q \in body^{-}(r)\}$$

$$= body^{-}(r)(R_{\Pi}(X,Y))$$

By $Y = S_{\Pi^{(X,Y)}}(X)$, we obtain $Y = body^{-}(r)(R_{\Pi}(X,Y))$.

Proof 4.2 Let Π be a logic program and let X and Y be two sets of atoms.
" => " : Let X be a justified answer set of Π with respect to Y and $\Pi' = R_{\Pi}(X,Y)$. By Theorem 4.9, we have $X = Cn^{+}(R_{\Pi}(X,Y)) = Cn^{+}(\Pi')$ and $Y = body^{-}(R_{\Pi}(X,Y)) = body^{-}(\Pi')$. We have to show that for each $r \in \Pi$

1. *If* $r \in \Pi'$, *then* $body^{+}(r) \subseteq X$ *and* $(X \cup \{head(r)\}) \cap (Y \cup body^{-}(r)) = \emptyset$.
2. *If* $r \notin \Pi'$, *then* $body^{+}(r) \nsubseteq X$ *or* $(X \cup \{head(r)\}) \cap (Y \cup body^{-}(r)) \neq \emptyset$.

Let $r \in \Pi$, then by Definition 4.1, we have $r \in \Pi'$ if and only if $body^{+}(r) \subseteq X$ *and* $(X \cup \{head(r)\}) \cap (Y \cup body^{-}(r)) = \emptyset$. Thus, the two conditions (1) and (2) hold.
" <= " : Let Π', $X = Cn^{+}(\Pi')$ and $Y = body^{-}(\Pi')$ such that for each $r \in \Pi$:

1. *If* $r \in \Pi'$, *then* $body^{+}(r) \subseteq X$ *and* $(X \cup \{head(r)\}) \cap (Y \cup body^{-}(r)) = \emptyset$.
2. 2.*If* $r \notin \Pi'$, *then* $body^{+}(r) \nsubseteq X$ *or* $(X \cup \{head(r)\}) \cap (Y \cup body^{-}(r)) \neq \emptyset$

We have to show X is a justified answer set of Π with respect to Y . Since for each $r \in \Pi'$, we have $body^{+}(r) \subseteq X$ *and* $(X \cup \{head(r)\}) \cap (Y \cup body^{-}(r)) = \emptyset$. It implies $\Pi' = R_{\Pi}(X,Y)$. Thus, we have $X = Cn^{+}(R_{\Pi}(X,Y)) = Cn^{+}(\Pi')$ and $Y = body^{-}(R_{\Pi}(X,Y)) = body^{-}(\Pi')$. Hence, by Theorem 4.1, we obtain X is a justified answer set of Π with respect to Y .

Proof 4.3 Let Π be a logic program and X be a set of atoms.
" => " : Let X be a justified answer set of Π. Consider $\Pi' = R_{\Pi}(X,Y)$. First, we have to show that

1. $body^{+}(\Pi') \subseteq Cn^{+}(\Pi')$
2. $body^{-}(\Pi') \cap Cn^{+}(\Pi') = \emptyset$.

From Theorem 4.1, we have $X = Cn^{+}(\Pi')$ and $Y = body^{-}(\Pi')$. By Definition 4.1, we have for each $r \in \Pi'$, *then* $body^{+}(r) \subseteq X$ *and* $(X \cup \{head(r)\}) \cap (Y \cup body^{-}(r)) = \emptyset$.
Thus, we obtain for each $\in \Pi'$ $body^{-}(r) \cap X = \emptyset$. By $X = Cn^{+}(\Pi')$, we have $body^{+}(\Pi') \subseteq Cn^{+}(\Pi')$ and $body^{-}(\Pi') \cap Cn^{+}(\Pi') = \emptyset$. Hence, Π' satisfies the two conditions (1) and (2).
Second, we show that $\Pi' = R_{\Pi}(X,Y)$ is maximal. Assume that Π' is not maximal satisfying the conditions

1. $body^{+}(\Pi') \subseteq Cn^{+}(\Pi')$
2. $body^{-}(\Pi') \cap Cn^{+}(\Pi') = \emptyset$.

That is, there is a strict superset Π'' for which the above conditions (1) and (2) hold and $X = Cn^{+}(\Pi'')$.
Hence, we have $body^{-}(\Pi'') \cap Cn^{+}(\Pi'') = \emptyset$. Since $\Pi' \subseteq \Pi''$ then, there is a rule $r'' \in \Pi''$ such that $body^{+}(\Pi'') \subseteq Cn^{+}(\Pi'')$ and $body^{-}(\Pi'') \cap Cn^{+}(\Pi'') = \emptyset$. Since $r'' \notin \Pi' = R_{\Pi}(X,Y)$, then by Definition 4.5, we have two cases

1: $body^{+}(\Pi'') \nsubseteq Cn^{+}(\Pi'')$

2: $(X \cup \{head(r'')\}) \cap (Y \cup body^{-}(r'')) \neq \emptyset$.

In case 1, we obtain $body^{+}(\Pi'') \nsubseteq Cn^{+}(\Pi'')$ which is a contradiction to $body^{+}(\Pi'') \subseteq Cn^{+}(\Pi'')$. Since for each $r'' \in \Pi''$, we have $body^{+}(\Pi'') \nsubseteq Cn^{+}(\Pi'')$ and $body^{-}(\Pi'') \cap Cn^{+}(\Pi'') = \emptyset$. This implies $head(r'') \in X = Cn^{+}(\Pi'')$.). Then, we have $X \cup \{head(r'')\} = Cn^{+}(\Pi'')$. This, in case 2, we have

$Cn^{+}(\Pi'') \cap (Y \cup body^{-}(r'')) = \emptyset$, but this is a contradiction to $body^{-}(\Pi'') \cap Cn^{+}(\Pi'') = \emptyset$, and $Y \cap Cn^{+}(\Pi'') = \emptyset$,where $Y = body^{-}(\Pi')$. Hence, Π' is a maximal set with the desired properties.

" <= " : Let $X = Cn^{+}(\Pi')$ and $Y = body^{-}(\Pi')$ for some maximal $\Pi' \subseteq \Pi$ such that $body^{+}(\Pi') \subseteq Cn^{+}(\Pi)$ and $body^{-}(\Pi') \cap Cn^{+}(\Pi') = \emptyset$. We have to show that X is a justified answer set of Π with respect to . For this it is sufficient to prove that Π' satisfies the two conditions (1) and (2) of Theorem 4.2. That is,

1. $If\ r \in \Pi', then\ body^{+}(r) \subseteq X\ and\ (X \cup \{head(r)\}) \cap (Y \cup body^{-}(r)) = \emptyset$.
2. $If\ r \notin \Pi', then\ body^{+}(r) \nsubseteq X\ or (X \cup \{head(r)\}) \cap (Y \cup body^{-}(r)) \neq \emptyset$.

We assume that the two conditions (1) and (2) of Theorem 4.2 is not satisfied. Then, there is some $r \in \Pi'$ such that

$(a)\ body^{+}(r) \nsubseteq X$

$(b)\ (X \cup \{head(r)\}) \cap (Y \cup body^{-}(r)) \neq \emptyset$.

In case (a), we have $body^{+}(r) \nsubseteq X = Cn^{+}(\Pi')$ which is a contradiction to $ody^{+}(r) \subseteq Cn^{+}(\Pi')$. Since for each $r \in \Pi'$, we have $body^{+}(r) \subseteq X$ and $body^{-}(r) \cap X = \emptyset$. This implies $head(r) \in X = Cn^{+}(\Pi')$. Then, we have $X \cup head(r) = Cn^{+}(\Pi')$. Thus in case (b), we have $Cn^{+}(\Pi') \cap (Y \cup body^{-}(r)) \neq \emptyset$. But this is a contradiction to $body^{-}(\Pi') \cap Cn^{+}(\Pi') = \emptyset$, where $body^{-}(r)) \subseteq Y = body^{-}(\Pi')$. Therefore, condition (1) of Theorem 4.2 is satisfied.

Now, we assume that the condition (2) of Theorem 4.2, is not satisfied. Then, there is some $r \notin \Pi'$ such that $body^{+}(r) \subseteq X\ and\ (X \cup \{head(r)\}) \cap (Y \cup body^{-}(r)) = \emptyset$. Thus, by Definition 4.1, we obtain $r \in R_{\Pi}(X, Y)$. Then $r \in \Pi'$, because otherwise Π' would not be a maximal subset of Π. But this is a contradiction to $r \notin \Pi'$. Hence, the condition (2) of Theorem 4.3, is satisfied.

References

[1] Baral, C. 2003. Knowledge representation, reasoning and declarative problem solving. Cambridge University Press.

[2] Bidoit, N., and Froidevaux, C. 1987. Minimalism subsumes default logic and circumscription in stratified logic programming. 89–97. IEEE Computer Society.

[3] Bidoit, N., and Froidevaux, C. 1991. Negation by default and unstratifiable logic programs. Theoretical Computer Science 78(1):85–112.

[4] Clark, K. 1978. Negation as failure. In Gallaire, H., and Minker, J., eds., Logic and Data Bases. Plenum Press. 293– 322.

[5] Delgrande, J.; Gharib, M.; Mercer, R.; Risch, V.; and Schaub, T. 2003. Łukaszewicz-style answer set programming: A preliminary report. In De Vos, M., and Provetti, A., eds., Proceedings of the Second International Workshop on Answer Set Programming (ASP'03), volume 78 of CEURWorkshop Proceedings.

[6] Dix, J. 1991. Classifying semantics of logic programs. In A. Nerode, W. M., and Subrahmanian, V. S., eds., Proceedings of the first International Workshop on logic Programming and Non-monotonic Reasoning, 166–180.

[7] Gebser, M.; Gharib, M.; Mercer, R. E.; and Schaub, T. 2009. Monotonic answer set programming. Journal of Logic and Computation 19(4):539 -564.

[8] Gebser, M.; Gharib, M.; and Schaub, T. 2007. Incremental answer sets and their computation. In Costantini, S., and Watson, R., eds., Proceedings of the Fourth International Workshop on Answer Set Programming (ASP'07), 161–174.

[9] Gelfond, M., and Lifschitz, V. 1991. Classical negation in logic programs and deductive databases. 9:365–385. Gharib, M.; Mercer, R.; and Schaub, T. 2007. Incremental answer set programming: A preliminary report. In Costantini, S., and Watson, R., eds., Proceedings of the Fourth International Workshop on Answer Set Programming (ASP'07), 149–160.

[10] Lifschitz, V. 1996. Foundations of logic programming. In Brewka, G.,ed., Principles of Knowledge Representation. CSLI Publications. 69–127.

[11] Lifschitz, V. 2002. Answer set programming and plan generation. Artificial Intelligence 138(1-2):39–54.

[12] Lloyd, J. 1987. Foundations of Logic Programming. Springer-Verlag.

[13] Łukaszewicz,W. 1988. Considerations on default logic: An alternative approach. Computational Intelligence 4:1–16. Makinson, D. 1989. General theory of cumulative inference. In Reinfrank, M.; de Kleer, J.; Ginsberg, M.; and Sandewall, E., eds., Proceedings of the Second International Workshop on Non-Monotonic Reasoning, volume 346, 1–

[14] Marek, V., and Truszczy´nski, M. 1999. Stable models and an alternative logic programming paradigm. In Apt, K.; Marek, W.; Truszczy´nski, M.; and Warren, D., eds., The Logic Programming Paradigm: a 25-Year Perspective. 375–398.

[15] Niemel¨a, I. 1999. Logic programs with stable model semantics as a constraint programming paradigm. Annals of Mathematics and Artificial Intelligence 25(3,4):241–273. Poole, D. 1991. The effect of knowledge on belief: conditioning, specificity, and the lottery paradox in default reasoning.Artificial Intelligence 49:281–307.

[16] Reiter, R. 1980. A logic for default reasoning. Artificial Intelligence 13(1-2):81–132.

[17] Tarski, A. 1955. A lattice-theoretic fixpoint theorem and its applications. Pacific Journal of Mathematics 5:285–309. Turner, H. 2001. Order-consistent programs are cautiously monotonic. 1(4):487–495.