RESEARCH PAPER | Computer Science | Volume : 3 | Issue : 8 | Aug 2013 | ISSN - 2249-555X



Stable Answer Set Programming

KEYWORDS: Answer set programming, Logic programs, Monotonicity, Applied rules, SLD resolution.

Mona Gharib
Mathematics Department, Faculty of Science, Zagazig University, Zagazig, Egypt

Amr I. Hassan
Mathematics and Computer Science Department, Faculty of Science, Port Said University, Port Said, 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 (ι-answer sets) have been introduced. The monotonicity of ι-answer sets is problematic, since it equates monotonic and nonmonotonic inferences. In this paper, we modify the definition of ι-answer sets in order to overcome these problems. Namely, we introduce the concept of stable answer sets which are characterized by their applied rules. Furthermore, we develop a sound and complete theorem proving method for our approach to answer set programming, which allows for query-oriented computations.*

1 INTRODUCTION

Answer Set Programming (ASP; [1]) was originally conceived as a declarative semantics for logic programming [8]. This is achieved by a bottom-up approach that is free of any procedural flavor. However, now that ASP has become an attractive tool for knowledge representation and reasoning, it is sometimes desirable to have top-down constructions that allow for query-answering (see eg. [2] for an application motivated approach). However, classical ASP and query-orientation appear to be completely diametrical. As detailed in [4], ASP does not allow or a goal-oriented computation. Related to this is the fact that answer set semanticists neither relevant nor modular. The principle of relevance states that the truth value of an atom only depends on the subprogram connected to this atom in the underlying dependency graph [5]. Accordingly, modularity stipulates that the semantics of the overall program can be composed of the semantics of subprograms (connected in the dependency graph [5]). In order to remedy these problems, an incremental answer set (ι -answer set) has been proposed in [3, 10, 7, 6, 9]. However, a ι-answer set may lack characteristic properties of classical answer sets, such as begin a model of the given program. Also, the monotonicity of ι-answer sets is problematic, since it equates monotonic and nonmonotonic inferences.

In this paper, we refine the definition of ι -answer sets in order to overcome these problems. Namely, we introduce the concept of stable answer sets which, can be determined one by one in the construction of stable answer set based on applicable rules. It guarantees the existence of stable answer sets for every logic program and facilitates their computation. We introduce a sound and complete theorem proving method for our approach to ASP, which allows for query-oriented computations. The outline of the paper is as follows: In Section 3, we introduce stable answer sets and in Section 4, we elaborate upon their formal properties. Section 5 and Section 6 deal with the relationships to ι-answer sets and classical answer sets of logic programs, respectively. In Section 7 and Section 8, we introduce a proof and derivation procedure for stable answer sets, respectively. Finally, we conclude with Section 9.

2 BACKGROUND

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

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

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

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 $body^+(\Pi) = \bigcup_{r \in \Pi} body^+(r)$ and $body^-(\Pi) =$

Volume : 3 | Issue : 8 | Aug 2013 | ISSN - 2249-555X

$\cup_{r\in\Pi} body^{-}(r)$. A literal is either an atom or a negated atom. A program Π 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) ∈ X whenever $body^{+}(r) \subseteq$ X. The smallest set of atoms which is closed under a basic program Π is denoted by $Cn(\Pi)$. The redact of a logic program Π 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 Π if $X = Cn(\Pi^{X})$.

For a program Π and a set X of atoms, the generating rules of X for Π are $\mathcal{G}_{\Pi}(X) = \{r \in \Pi \mid body^{+}(r) \subseteq X \ and \ body^{-}(r) \cap X = \emptyset\}$.

In fact, one can show that X is an answer set of Π iff $X = Cn^{+}(\mathcal{G}_{\Pi}(X)) = Cn(\Pi^{X}) = X$. Furthermore, any answer set of a program Π is also a model of Π: A set X of atoms is a model of Π if, for all $r \in \Pi$, we have $head(r) \in X$ if $body^{+}(r) \subseteq X \ and \ body^{-}(r) \cap X = \emptyset$.

In [6], a ι-answer set of a logic program was defined as follows: A set X of atoms is a ι-answer set of a logic program Π iff $X = Cn(\Pi')$ for some ⊆-maximal $\Pi' \subseteq \Pi$, such that

1. $body^{+}(\Pi') \subseteq Cn(\Pi') \ and$

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

For example, consider the following logic program Π_1:

$r_1 : a \leftarrow$

$r_2 : b \leftarrow not \ a$ (2)

The ι-answer sets of Π_1 are $\{a\}$ and $\{b\}$.

3 STABLE ANSWER SETS

The construction of answer sets is non-modular, that is, all rules of a program need to be inspected. It is thus impossible to incrementally construct an answer set or to locally validate a construction, like a proof, by only looking at a subset of the given program. In [10], they introduced ι-answer sets to overcome these problems, where ι indicates the incremental flavor. But the monotonicity of ι-answer sets equates monotonic and nonmonotonic inferences. To see this, let us reconsider program Π_1 in (2). This program has two ι-answer sets $\{a\}$ and $\{b\}$. As we see, the monotonic inference of a does not override the nonmonotonic inference of b to eliminate the second ι-answer set $\{b\}$ of Π_1.

Thus, we modify the definition of ι-answer sets to remedy these problems. This leads us to stable answer sets.

Definition 3.1 Let Π be a logic program.

A set X of atoms is a stable answer set of Π iff $X = Cn(\Pi')$ for some ⊆-maximal $\Pi' \subseteq \Pi$, such that

1. $body^{+}(\Pi') \subseteq Cn(\Pi')$,

2. $body^{-}(\Pi') \cap Cn(\Pi') = \emptyset, and$

3. $body^{-}(\Pi') \cap Cn(\Pi^{X}) = \emptyset$.

Definition 3.1 characterizes a stable answer set X of Π in terms of the rules that apply with respect to X. The set Π' of such rules is maximal among all subsets of Π that satisfy Conditions (1)–(3). Condition (1) guarantees that the positive bodies of rules in Π' are justified, and Condition (2) makes sure that the rules in Π' do not block one another, while in Condition (3) non-applied rules are additionally taken into account. The difference between Definition of ι-answer sets and Definition 3.1 of stable answer sets lies in the third condition. In definition ι -answer, Condition (2) makes sure that the applied rules do not block each other, while in Definition3.1; non-applied rules are additionally taken into account. In the following, we show how the existence of Condition (3) in Definition 3.1 modifies the definition of ι-answer sets.

Theorem 3.1 Given the same prerequisites as in Definition 3.1.

Then, Condition (2) and (3) in Definition 3.1 are equivalent to

(2′) $body^{-}(\Pi') \cap Cn(\Pi' \cup (\Pi \setminus \Pi')^{X}) = \emptyset$.

In fact, Condition (3) in Definition 3.1 deals with both rules in Π' as well as those in $(\Pi \setminus \Pi')^{X}$. Condition (2′) is obtained from Condition (2) in the definition of ι-answer sets, stipulating that $body^{-}(\Pi') \cap Cn(\Pi') = \emptyset$, by adding $(\Pi \setminus \Pi')^{X}$. The purpose of the reduct with respect to X is to eliminate all rules in $(\Pi \setminus \Pi')^{X}$ that are blocked when applying the rules in Π'. Note that

Volume : 3 | Issue : 8 | Aug 2013 | ISSN - 2249-555X

Condition (2′) in Theorem 3.1 assures that no rule blocks another in Π' and also that no potentially applicable rule in $(\Pi \setminus \Pi')^X$ blocks any rule in Π'. That is, for each $r \in (\Pi \setminus \Pi')$ such that $body^+(r) \subseteq X$ and $body^-(r) \cap X = \emptyset$, we have $head(r) \notin body^-(\Pi')$.

The modification shown in Theorem 3.1 has an important effect on the formal properties of stable answer sets. To see this, let us reconsider program Π_1 in (2), consisting of rules:

$r_1 : a \leftarrow$

$r_2 : b \leftarrow not\ a$

This program has $\{a\}$ as its unique stable answer set, which is also the unique classical answer set of Π_1. For $\{a\}$, the corresponding maximal subset of Π_1 is

$\Pi_1' = \{r_1\}$ We have

$body^+(\Pi_1') = \emptyset \subseteq \{a\} = Cn(\Pi_1')$,

$body^-(\Pi_1') \cap Cn(\Pi_1') = \emptyset \cap \{a\} = \emptyset$, and

$body^-(\Pi_1') \cap Cn(\Pi_1^{\{a\}}) = \emptyset \cap \{a\} = \emptyset$.

Since
$body^-(r_2) \cap Cn(\Pi_1^{\{a\}}) = \{a\} \cap \{a\} \neq \emptyset$, is indeed a maximal subset of Π_1 such that Conditions (1), (2), and (3) in Definition 3.1 hold. Since, Π_1 has two ι-answer sets, $\{a\}$ and $\{b\}$, which is arguably counterintuitive, because the monotonic inference of a cannot override the nonmonotonic inference of b to eliminate the second ι-answer set $\{b\}$. For $\{b\}$, the corresponding maximal subset of Π_1 is $\Pi_1'' = \{r_2\}$. We have

$body^+(\Pi_1'') = \emptyset \subseteq \{b\} = Cn(\Pi_1'')$,

$body^-(\Pi_1'') \cap Cn(\Pi_1'') = \{a\} \cap \{b\} = \emptyset$, and

$body^-(\Pi_1'') \cap Cn(\Pi_1^{\{b\}}) = \{a\} \cap \{a, b\} = \{a\} \neq \emptyset$.

Thus, Condition (3) in Definition 3.1 does not hold for Π_1''. Indeed, Π_1 has $\{a\}$ as its unique stable answer set.

4 BASIC PROPERTIES OF STABLE ANSWER SETS

In this section, we provide some properties of logic programs and their stable answer sets, and we compare stable answer sets to ι -answer sets and classical answer sets. Based on the concept of generating rules, we can obtain stable answer sets as follows.

Theorem 4.1 Let Π be a logic program and X be a set of atoms.

If $X = Cn(\mathcal{G}_\Pi(X))$, then X is a stable answer set of Π.

For illustration, consider the following logic program Π_2 consisting of rules:

$r_1 : a \leftarrow not\ b$

$r_2 : b \leftarrow not\ a$ (3)

For $\{a\}$, we have $\mathcal{G}_{\Pi_2}(\{a\}) = \{r_1\}$ and $Cn\left(\mathcal{G}_{\Pi_2}(\{a\})\right) = \{a\}$. That is, $\{a\}$ is a stable answer set of Π_2. For $\{b\}$, we have $\mathcal{G}_{\Pi_2}(\{b\}) = \{r_2\}$ and $Cn\left(\mathcal{G}_{\Pi_2}(\{b\})\right) = \{b\}$. Then, $\{b\}$ is a stable answer set of Π_2. Thus, Π_2 has two stable answer sets $\{a\}$ and $\{b\}$, which are identical to its classical answer sets and ι-answer sets.

Unlike classical answer sets, the concept of generating rules does not provide an alternative characterization of stable answer sets. This leads to the concept of *applied rules*.

Definition 4.1 Let Π be a logic program and X be a set of atoms.

We define the applied rules of X for Π as

$\mathcal{A}_\Pi(X) = \{r \in \Pi \mid body^+(r) \subseteq X,\ body^-(r) \cap X = \emptyset, head(r) \in X\}$.

Note that $\mathcal{A}_\Pi(X) \subseteq \mathcal{G}_\Pi(X)$ for any program Π and any set X of atoms, but not vice versa. In view of Definition 3.1, we derive the following result.

Theorem 4.2 Let Π be a logic program and X be a set of atoms.

If X is a stable answer set of Π, then $\mathcal{A}_\Pi(X)$ is a ⊆maximal subset of Π such That

1. $body^+(\mathcal{A}_\Pi(X)) \subseteq Cn(\mathcal{A}_\Pi(X))$,

2. $body^-(\mathcal{A}_\Pi(X)) \cap Cn(\mathcal{A}_\Pi(X)) = \emptyset, and$

Volume : 3 | Issue : 8 | Aug 2013 | ISSN - 2249-555X

3. $body^{-}\left(\mathcal{A}_{\Pi}(X)\right) \cap Cn(\Pi^{X}) = \emptyset$.

That is, for a stable answer set X of Π, the applied rules $\mathcal{A}_{\Pi}(X)$ correspond to Π_1' in Definition 3.1.

As a consequence of Definition 3.1 and Theorem 4.2, we obtain the following result.

Corollary 4.3 Let Π be a logic program and X be a set of atoms.

If X is a stable answer set of Π, then $X = Cn(\mathcal{A}_{\Pi}(X))$.

The following result shows that stable answer sets can be constructed, as the corresponding applied rules underlie monotonicity.

A strategy would be to start with a set of applied rules Π' of a set X' of atoms, then there is a stable answer set X of Π with $X' \subseteq X$, provided that $body^{-}(\Pi') \cap Cn\left(\Pi^{X'}\right) = \emptyset$. This is made precise in the following theorem.

Theorem 4.4 Let Π be a logic program and $\Pi' \subseteq \Pi$.

If Π' has a stable answer set X' and $body^{-}(\Pi') \cap Cn\left(\Pi^{X'}\right) = \emptyset$, then Π has a stable answer set X such that $X' \subseteq X$.

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

$r_1 : a \leftarrow$

$r_2 : b \leftarrow not\ a$

For $\Pi_1' = \{r_2\}$, one can verify that $\{b\}$ is a stable answer set of Π_1'. But, we have

$body^{-}(\Pi_1') \cap Cn\left(\Pi_1^{\{b\}}\right) = \{a\} \cap \{a, b\} = \{a\} \neq \emptyset$.

Thus, $\{b\}$ cannot be extended to a stable answer set of Π_1. It is easy to see that $\{a\}$ is a stable answer set of $\Pi_1'' = \{r_1\}$. We have $body^{-}(\Pi_1'') \cap Cn\left(\Pi_1^{\{a\}}\right) = \emptyset \cap \{a\} = \emptyset$. Then, $\{a\}$ can be extended to a stable answer set of Π_1. Observe that r_2 cannot be added to Π_1'' without violating Condition (3) in Definition 3.1. Indeed, $\{a\}$ is the unique stable answer set of Π_1. The previous result can be expressed in terms of applied rules as follows.

Consequently, We conclude the following result for stable answer sets as follows.

Corollary 4.5 Every logic program has a stable answer set.

For illustration, consider the following logic program Π_3 consisting of rules:

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

$$r_2 : b \leftarrow not\ c, \qquad (4)$$

$$r_3 : c \leftarrow not\ a.$$

This program has $\emptyset$ as its unique stable answer set, while it has no classical answer sets.

Corollary 4.6 Let Π be a logic program and $\Pi' \subseteq \Pi$.

If Π' satisfies Conditions (1)–(3) in Definition 3.1, then there exists a stable answer set X of Π such that $\Pi' \subseteq \mathcal{A}_{\Pi}(X)$.

5 RELATIONSHIP TO ANSWER SET SEMANTICS

In this section, we further discuss the relationship between classical answer sets and stable answer sets of logic programs.

Theorem 5.1 Let Π be a logic program and X be a set of atoms.

If X is an answer set of Π, then X is a stable answer set of Π.

That is, the set of stable answer sets of a logic program forms a superset of its classical answer sets.

Consider the following logic program Π_4 :

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

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

$$r_3 : p \leftarrow not\ a \qquad (5)$$

$$r_4 : p \leftarrow not\ p$$

This program has two stable answer sets: $\{b, p\}$ and $\{a\}$ but $\{b, p\}$ is the only classical answer set of Π_4.

Volume : 3 | Issue : 8 | Aug 2013 | ISSN - 2249-555X

A stable answer set is also an answer set if its applied and generating rules are identical.

Theorem 5.2 Let Π be a logic program and X be a stable answer set of Π.

Then, X is an answer set of Π iff $\mathcal{A}_{\Pi}(X) = \mathcal{G}_{\Pi}(X)$.

Based on the concept of models, we can reformulate Theorem 5.2 as follows.

Theorem 5.3 Let Π be a logic program and X be a stable answer set of Π.

Then, X is an answer set of Π iff X is a model of Π.

6 RELATIONSHIP TO ι-ANSWER SET SEMANTICS

In this Section, we further characterize the relationship between ι-answer sets and stable answer sets of logic programs.

Theorem 6.1 Let Π be a logic program and X be a ι-answer set of Π.

Then, X is a stable answer set of Π if $body^{-}\left(\mathcal{A}_{\Pi}(X) \cap Cn(\Pi^{X})\right) = \emptyset$.

In other words, a ι-answer set X of a program Π is a stable answer set if there is no rule r ∈ Π such that $body^{+}(r) \subseteq X$, $body^{-}(r) \cap X = \emptyset$, $head(r) \in body^{-}(\mathcal{A}_{\Pi}(X))$. Let us reconsider program Π_3 in (4), consisting of rules:

$r_1 : a \leftarrow not\ b$

$r_2 : b \leftarrow not\ c$

$r_3 : c \leftarrow not\ a$

We have seen that Π_3 has three ι-answer sets: $\{a\}$, $\{b\}$ and $\{c\}$. For $\{a\}$, we have

$\mathcal{A}_{\Pi}(\{a\}) = \{r_1\}$. Then, $body^{-}\left(\mathcal{A}_{\Pi_3}(\{a\})\right) \cap Cn\left(\Pi_3^{\{a\}}\right) = \{b\} \cap \{a, b\} \neq \emptyset$ .Thus, $\{a\}$ is not a stable answer set of Π_3. For the other -answer sets, $\{b\}$ and $\{c\}$. One can similarly verify that they are no stable answer sets of Π_3. Indeed, Π_3 has ø as its unique stable answer set, while Π_3 has no classical answer sets.

Since, the concept of applied rules yields an alternative characterization of ι-answer sets [10, 9]. Then, from this characterization and Corollary 4.3, we conclude the following result.

Theorem 6.2 Let Π be a logic program and X be a stable answer set of Π.

Then, X is a ι-answer set of Π if for every $r \in \Pi \setminus \mathcal{A}_{\Pi}(X)$, either

1. $body^{+}(r) \not\subseteq X$,

2. $body^{-}(r) \cap X \neq \emptyset, or$

3. $head(r) \in body^{-}(\mathcal{A}_{\Pi}(X) \cup \{r\})$.

7 QUERY-ANSWERING

In this section, we consider the following problem: Given a program Π and an atom p, determine where p belongs to some stable answer set of Π. To this end, we introduce the concept of stable proofs.

Definition 7.1 Let Π be a logic program and p be an atom. We define Π' as a stable proof for p from Π iff $\Pi' \subseteq \Pi$ such that

1. $p \in head(\Pi')$,

2. $body^{+}(\Pi') \subseteq Cn(\Pi')$,

3. $body^{-}(\Pi') \cap Cn(\Pi') = \emptyset\ and$

4. $body^{-}(\Pi') \cap Cn(\Pi^{head(\Pi')}) = \emptyset$.

In Definition 7.1, non-applied rules are additionally taken into account, it lies in the fourth condition. That is, any stable proof for an atom p consists of rules allowed for deriving p.

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

$r_1 : a \leftarrow$

$r_2 : b \leftarrow not\ a$

The atom a has the stable proof $\Pi' = \{r_1\}$ from Π_1 . We have

$$a \in head(\Pi_1')$$

$$body^{+}(\Pi_1') = \emptyset \subseteq \{a\} = Cn(\Pi_1'),$$

Volume : 3 | Issue : 8 | Aug 2013 | ISSN - 2249-555X

$body^-(\Pi_1') \cap Cn(\Pi_1') = \emptyset \cap \{a\} = \emptyset$, and

$body^-(\Pi_1') \cap Cn\left(\Pi_1^{\{a\}}\right) = \emptyset \cap \{a\} = \emptyset.$

While, the other atom b has no stable proof from Π_1. For b and $\Pi_1'' = \{r_2\}$ we have

$$b \in head(\Pi_1'')$$

$$body^+(\Pi_1'') = \emptyset \subseteq \{b\} = Cn(\Pi_1''),$$

$$body^-(\Pi_1'') \cap Cn(\Pi_1'') = \{a\} \cap \{b\} = \emptyset, \text{ and}$$

$$body^-(\Pi_1'') \cap Cn(\Pi_1^{\{b\}}) = \{a\} \cap \{a,b\} = \{a\} \neq \emptyset.$$

Thus, condition (4) in Definition 7.1 does not hold for $\Pi_1'' = \{r_2\}$. Indeed, b has no stable proof from Π_1

In Corollary 4.6, we have shown that for a program Π and $\Pi' \subseteq \Pi$, there exists a stable answer set X of Π such that $\Pi' \subseteq \mathcal{A}_\Pi(X)$, if Π' satisfies Conditions (1)–(3) in Definition 3.1. Thus, we have the following result.

Theorem 7.1 Let Π be a logic program and p be an atom.

Then, p has a stable proof from Π iff $p \in X$ for some stable answer set X of Π.

That is, the existence of a stable proof for an atom p from a program Π guarantees the existence of a stable answer set of Π containing p.

For example, consider program $\Pi_5 = \{a \leftarrow not\ a\}$. According to Definition 7.1, a has no stable proof from Π_4. Thus, Π_4 has no stable answer set containing a. Indeed, we have seen that Π_4 has $\emptyset$ as its unique stable answer set.

8 DERIVATION PROCEDURE

Now, we introduce a calculus along with the concept of a derivation such that, given a logic program Π and a goal g, the derivation succeeds iff g holds with respect to a Stable answer set of Π. A goal g is of the form $\leftarrow p_1, \ldots, p_n$ where p_i $(1 \leq i \leq n)$ is an atom. A context C is a set of literals.

Definition 8.1 Let g be a goal and C be a context. We define the ordered pair (g, C) as a contextual goal; g is said to be a goal in context C.

We denote the set of positive atoms in C by C^+ and the set of atoms preceded by not by C^-. A selection function f is a function from a set of goals to a set of atoms that maps a goal to one of its contained atoms. In the following, we introduce a proof procedure analogous to SLD resolution [13].

Definition 8.2 Let Π be a logic program, f be a selection function, and (g, C) be a contextual goal.

Then, a contextual goal (g', C') is derivable from (g, C) and a rule $r \in \Pi$ via f, if the following conditions hold:

1. $p = f(g)$,
2. $p = head(r)$,
3. $g' = (body(g) \setminus \{p\}) \cup body^+(r)$,
4. $C' = (C \cup body^-(r) \cup \{p\})$,
5. $C' \cap Cn(\Pi^{body(g)}) = \emptyset$, *and*
6. $C'^+ \cap C'^- = \emptyset$.

Definition 8.3 Let Π be a logic program, f be a selection function, and (g, C) be a contextual goal.

A stable derivation of g from Π in context C via f consists of a finite sequence $\langle (g_0, C_0), \ldots, (g_n, C_n) \rangle$ of contextual goals and a finite sequence of rules $\langle r_1, \ldots, r_n \rangle$ such that each (g_{i+1}, C_{i+1}) is derived from (g_i, C_i) and $r_{i+1} \in \Pi$ via f, where $0 \leq i < n$ and $(g_0, C_0) = (g, C)$.

A successful stable derivation is one that ends in an ordered pair with an empty goal $(i.e. body(g) = \emptyset)$, otherwise, it is called a failed stable derivation. We refer to the last ordered pair in a successful stable derivation of g from Π in context C

via f by $(\square, C_\square)$, where $C_\square$ is the projective mapping

$C_\square : \langle (g_0, C_0), \ldots, (\square, C_n) \rangle \longrightarrow C_n$.

For illustration, consider the following logic program Π_7 :

$r_1 : p \leftarrow$

$r_2 : q \leftarrow p, not\ r$

$r_3 : q \leftarrow r, not\ p$ (5)

Volume : 3 | Issue : 8 | Aug 2013 | ISSN - 2249-555X

$r_4 : r \leftarrow p, not\ s$

This program has $\{p, r\}$ as the only stable answer set. A successful stable derivation of $\leftarrow r$ from Π_7 in context $\emptyset$ is:

(1) $(\leftarrow r, \emptyset)$

(2) $(\leftarrow p, \{r, not\ s\})$ $derived\ from$ (1) $and\ r_4$

(3) $(\leftarrow \square, \{p, r, not\ s\})$ $derived\ from$ (2) $and\ r_1$

$success.$

A failed stable derivation of $\leftarrow q$ from Π_7 in context $\emptyset$ is:

(1) $(\leftarrow q, \emptyset)$

(2) $(\leftarrow p, \{q, not\ r\})$ $derived\ from$ (1) $and\ r_2$

$failur.$

Because the derivation of $\leftarrow p$ from Π_7 in context $\{q, not\ r\}$ fails to satisfy Condition (5) in Definition 8.2.

The following result demonstrates the soundness and completeness of our stable derivation method.

Theorem 8.1 Let Π be a logic program, f be a selection function, and p be an atom.

Then, there is a stable proof for p from Π iff there is a successful stable derivation of $\leftarrow p$ from Π in context $\emptyset$ via f.

The following result is obtained as consequences of Theorem 7.1 and 8.1.

Corollary 8.2 Let Π be a logic program, f be a selection function, and p be an atom.

Then, there is a successful stable derivation of $\leftarrow p$ from Π in context $\emptyset$ via f iff $p \in X$ for some stable answer set of Π.

9 CONCLUSION

In this work, we have introduced stable answer sets as another alternative approach to answer set programming. The fundamental idea was to modify the definition of ι-answer sets in a way re-establishing some properties of answer sets, such as monotonicity property [3, 10, 7, 6, 9]. We have shown that, every logic program has at least one stable answer set, which can be constructed incrementally based on applicable rules. Furthermore, we have investigated the relationship between stable answer sets and classical answer sets of logic programs. The stable answer sets of a logic program form a superset of its (standard) answer sets. Based on the concept of models, we have shown that a stable answer set that is a model of the given program is also an answer set. A Stable answer set is an answer set if its applied and generating rules are identical.

We have furnished a theory for answer set programming that provides local computations via top-down query answering. We have developed a sound and complete proof procedure for our incremental approach to answer set programming that allows for query-oriented computations. Important properties and advantages of our procedure are: It can be used to decide whether an atom belongs to some stable answer set of a program.

A Proofs

This section presents the proofs of the results in paper.

Theorem A.1 Let Π be a logic program and X be a set of atoms. Then X is an answer set of Π iff $X = Cn(\mathcal{G}_\Pi(X))$.

Proof A.1 see the proof in [11].

Proof 3.1 Let Π be a logic program, $\Pi' \subseteq \Pi$ such that $X = Cn(\Pi')$ and $body^+(\Pi') \subseteq Cn(\Pi')$. We have to prove that $body^-(\Pi') \cap Cn(\Pi' \cup (\Pi \setminus \Pi')^X) = \emptyset$ iff $body^-(\Pi') \cap Cn(\Pi') = \emptyset$, and $body^-(\Pi') \cap Cn(\Pi^X) = \emptyset$.

" ➔ " : Let $body^-(\Pi') \cap Cn(\Pi' \cup (\Pi \setminus \Pi')^X) = \emptyset$. Then, we conclude that $body^-(\Pi') \cap Cn(\Pi') = \emptyset$. It remains to show that $y^-(\Pi') \cap Cn(\Pi^X) = \emptyset$. We first show that $Cn(\Pi' \cup (\Pi \setminus \Pi')^X)$ is closed under Π^X. Let $r^+ \in \Pi^X$ such that $body^+(r) \subseteq Cn(\Pi' \cup (\Pi \setminus \Pi')^X)$. Then, we have either $r \in \Pi'$ or $r^+ \in (\Pi \setminus \Pi')^X$ so, that $head(r) \in Cn(\Pi' \cup (\Pi \setminus \Pi')^X)$. Hence, $Cn(\Pi' \cup (\Pi \setminus \Pi')^X)$ is closed under Π^X. Since $Cn(\Pi^X)$ is the smallest set being closed under Π^X, we have that $Cn(\Pi^X) \subseteq Cn(\Pi' \cup (\Pi \setminus \Pi')^X)$. From this and $body^-(\Pi') \cap Cn(\Pi' \cup (\Pi \setminus \Pi')^X) = \emptyset$, we have $body^-(\Pi') \cap Cn(\Pi^X) = \emptyset$.

Volume : 3 | Issue : 8 | Aug 2013 | ISSN - 2249-555X

"◀= ": Assume that $body^{-}(\Pi') \cap Cn(\Pi') = \emptyset$, and $body^{-}(\Pi') \cap Cn(\Pi^{X}) = \emptyset$. We have to show that $body^{-}(\Pi') \cap Cn(\Pi' \cup (\Pi \setminus \Pi')^{X}) = \emptyset$. Let $r \in (\Pi' \cup (\Pi \setminus \Pi')^{X})$. Then, we have either $r \in \Pi'$ or $r^{+} \in (\Pi \setminus \Pi')^{X}$. If $r \in \Pi'$, then by $body^{-}(\Pi') \cap Cn(\Pi^{X}) = \emptyset$ and $Cn(\Pi') = X$, we have $body^{-}(r) \cap X = \emptyset$. If $r^{+} \in (\Pi \setminus \Pi')^{X}$, then we have $body^{-}(r) \cap X = \emptyset$. Thus, we have that $r^{+} \in \Pi^{X}$. We conclude that $(\Pi' \cup (\Pi \setminus \Pi')^{X})^{\emptyset} \subseteq \Pi^{X}$. The monotonicity of Cn implies that $Cn(\Pi' \cup (\Pi \setminus \Pi')^{X})^{\emptyset} \subseteq Cn(\Pi^{X})$. Since $body^{-}(\Pi') \cap Cn(\Pi^{X}) = \emptyset$, we have that $body^{-}(\Pi') \cap Cn(\Pi' \cup (\Pi \setminus \Pi')^{X}) = \emptyset$.

Proof 4.1 Let Π be a logic program and X be a set of atoms. Let $X = Cn(\mathcal{G}_{\Pi}(X))$. We have to show that X is a stable answer set of Π. For each rule $r \in \mathcal{G}_{\Pi}(X)$ we have $body^{+}(r) \subseteq X$ and $body^{-}(X) \cap X = \emptyset$. Then, by $X = Cn(\mathcal{G}_{\Pi}(X))$, we have $body^{+}(\mathcal{G}_{\Pi}(X)) \subseteq Cn(\mathcal{G}_{\Pi}(X))$ and $body^{-}(\mathcal{G}_{\Pi}(X)) \cap Cn(\mathcal{G}_{\Pi}(X)) = \emptyset$.

Since $X = Cn(\mathcal{G}_{\Pi}(X))$, by Theorem A.1, we have that X is answer set of Π. Then, we have $X = Cn(\Pi^{X})$. We conclude that $X = Cn(\Pi^{X}) = Cn(\mathcal{G}_{\Pi}(X))$. By $body^{-}(\mathcal{G}_{\Pi}(X)) \cap Cn(\mathcal{G}_{\Pi}(X)) = \emptyset$, we have $body^{-}(\mathcal{G}_{\Pi}(X)) \cap Cn(\Pi^{X}) = \emptyset$. Thus, Conditions (1)-(3) in Definition 3.1 hold for $\Pi' = \mathcal{G}_{\Pi}(X)$. So, if we prove that $\mathcal{G}_{\Pi}(X)$ is maximal, then the proof of theorem is established. Consider some $r \in \mathcal{G}_{\Pi}(X)$). By the definition of $\mathcal{G}_{\Pi}(X)$, either $body^{+}(r) \nsubseteq X$ or $body^{-}(r) \cap X \neq \emptyset$. We conclude that for any Π'' such that $\mathcal{G}_{\Pi}(X) \subseteq \Pi''$, either $body^{+}(\Pi'') \nsubseteq Cn(\Pi'')$, $body^{-}(\Pi'') \cap Cn(\Pi'') \neq \emptyset$, or $body^{-}(\Pi'') \cap Cn(\Pi^{X}) \neq \emptyset$. Thus, $\Pi' = \mathcal{G}_{\Pi}(X)$ is a maximal subset of Π satisfying Conditions (1)-(3) in Definition 3.1.

Lemma A.1 Let Π be a logic program, $\Pi' \subseteq \Pi$ and X be a set of atoms. Then, $Cn(\Pi') \subseteq Cn(\Pi^{X})$, if $body^{-}(\Pi') \cap X = \emptyset$.

Proof A.1 Let, $body^{-}(\Pi') \cap X = \emptyset$. Then, for each $r \in \Pi'$, we have $body^{-}(r) \cap X = \emptyset$. By $\Pi' \subseteq \Pi$, we have $r^{+} = head(r) \leftarrow body^{+}(r) \in \Pi^{X}$. Then, $\Pi'^{\emptyset} \subseteq \Pi^{X}$, by the monotonicity of Cn, we have $Cn(\Pi'^{\emptyset}) \subseteq Cn(\Pi^{X})$. For any Program Π'', the definition of Cn implies that $Cn(\Pi''^{\emptyset}) = Cn(\Pi'')$. Then, we conclude that $Cn(\Pi') \subseteq Cn(\Pi^{X})$.

Lemma A.2 Let Π be a logic program and X be a stable answer set of Π. Then, $X \subseteq Cn(\Pi^{X})$.

Proof A.2 Let X be a stable answer set of Π. Then, by theorem 3.1, we have $X = Cn(\Pi')$. for some $\subseteq$-maximal $\Pi' \subseteq \Pi$ such that

1. $body^{+}(\Pi') \subseteq Cn(\Pi')$,

2. $body^{-}(\Pi') \cap Cn(\Pi' \cup (\Pi \setminus \Pi')^{X}) = \emptyset$.

We have to show that $X \subseteq Cn(\Pi^{X})$. From (3), we have $body^{-}(\Pi') \cap Cn(\Pi') = \emptyset$. By $X = Cn(\Pi')$, we have $body^{-}(\Pi') \cap X = \emptyset$. According to Lemma A.1, we have $Cn(\Pi') \subseteq Cn(\Pi^{X})$. From this and $X = Cn(\Pi')$, we have $X = Cn(\Pi^{X})$.

Proof 4.2 Let Π be a logic program and X be a set of atoms. Let X is a stable answer set of . Then, by Definition 3.1, we have $X = Cn(\Pi')$ for some $\subseteq$-maximal Π such that

1. $body^{+}(\Pi') \subseteq Cn(\Pi')$,

2. $body^{-}(\Pi') \cap Cn(\Pi') = \emptyset$ *and*

3. $body^{-}(\Pi') \cap Cn(\Pi^{X}) = \emptyset$.

We have to show that $\Pi' = \mathcal{A}_{\Pi}(X)$, where $\mathcal{A}_{\Pi}(X) = \{r \in \Pi \mid body^{+}(r) \subseteq X,\ body^{-}(r) \cap X = \emptyset$ *and* $head(r) \in X\}$

" $\subseteq$ " : Let $r \in \Pi'$. Then, by $X = Cn(\Pi')$ and (1), we have $body^{+}(r) \subseteq X$. As X is a stable answer set of Π, by Lemma A.2, we have $X \subseteq Cn(\Pi^{X})$. The latter and (2) imply that $body^{-}(\Pi') \cap X = \emptyset$. By $X = Cn(\Pi')$, we have $body^{-}(\Pi') \cap Cn(\Pi') = \emptyset$. Since $X = Cn(\Pi')$ is closed under Π', we have that $head(r) \in X$. From the definition $\mathcal{A}_{\Pi}(X)$, we have $r \in \mathcal{A}_{\Pi}(X)$, which implies that $\Pi' \subseteq \mathcal{A}_{\Pi}(X)$

"⊇.": Let $r \in \mathcal{A}_{\Pi}(X)$. Then, $body^{+}(r) \subseteq X$, $body^{-}(r) \cap X = \emptyset$ *and* $head(r) \in X$. As X is a stable answer set of Π, by Lemma A.2, we have $X \subseteq Cn(\Pi^{X})$. From this and (2), we have $body^{-}(\Pi') \cap Cn(\Pi') = \emptyset$. Since Π' is maximal subset in Π satisfying Conditions (1), (2), and (3), we have $r \in \Pi'$. We conclude that $\mathcal{A}_{\Pi}(X) \subseteq \Pi'$.

Volume : 3 | Issue : 8 | Aug 2013 | ISSN - 2249-555X

Proof 4.3 Let Π be a logic program and X be a stable answer set of Π. Then, by Theorem 4.2, we have $\Pi' = \mathcal{A}_\Pi(X)$, where $\mathcal{A}_\Pi(X)$ is maximal in Π satisfying Conditions (1)–(3) in Definition 3.1 . Thus, $Cn(\Pi') = Cn(\mathcal{A}_\Pi(X))$. By $X = Cn(\Pi')$, we have $X = Cn(\mathcal{A}_\Pi(X))$.

Proof 4.5 Let Π be a logic program and X be a set of atoms. Let $\Pi' = \emptyset$. Then, for $X = Cn(\Pi') = \emptyset$.

1. $body^+(\Pi') = \emptyset \subseteq Cn(\Pi')$,

2. $body^-(\Pi') = \emptyset \cap Cn(\Pi') = \emptyset$ *and*

3. $body^-(\Pi') = \emptyset \cap Cn(\Pi^X) = \emptyset$.

Thus, there exists $\Pi' \subseteq \Pi$ satisfying Conditions (1)–(3) in Definition 3.1. This implies that there is a $\subseteq$-maximal $\Pi' \subseteq \Pi$ satisfying Conditions (1)–(3) in Definition 3.1.

Proof 4.6 Let Π be a logic program and $\Pi' \subseteq \Pi$ satisfying Conditions (1)–(3) in Definition 3.1.

Clearly, there is some maximal $\Pi'' \subseteq \Pi$ such that $\Pi' \subseteq \Pi''$ and Conditions (1)–(3) in Definition 3.1 are satisfied for Π''. Then, $X = Cn(\Pi'')$ is a stable answer set of Π, and by Theorem 4.2, we have that $\Pi'' = \mathcal{A}_\Pi(X)$. Thus, $\Pi' \subseteq \mathcal{A}_\Pi(X)$.

Proof 5.1 Let Π be a logic program and X be an answer set of Π . We have to show that X is a stable answer set of Π. As X is an answer set of Π, by Theorem A.1, we have $X = Cn(\mathcal{G}_\Pi(X))$. Then, by Theorem 4.1, we have that X is a stable answer set of Π.

Lemma A.3 Let Π be a logic program and X be an answer set of Π. Then, X is a model of Π.

Proof A.3 Let X be an answer set of a logic program . Then, by Theorem A.1, we have $X = Cn(\mathcal{G}_\Pi(X))$. It follows that for each $r \in \Pi$ such that $body^+(r) \subseteq X$, *and* $body^-(r) \cap X = \emptyset$, we have $head(r) \in X$. Thus, X is a model of Π.

Proof 5.2 Let Π be a logic program and X be a stable answer set of Π.

" =▶" : Let X be an answer set of Π. Then by Lemma A.3, we have that X is a model of Π. That is, for every $r \in \Pi$ such that $body^+(r) \subseteq X$, *and* $body^-(r) \cap X = \emptyset$, we have $head(r) \in X$. This implies that $\mathcal{G}_\Pi(X) \subseteq \mathcal{A}_\Pi(X)$. From the definition of $\mathcal{A}_\Pi(X)$, we have $\mathcal{A}_\Pi(X) \subseteq \mathcal{G}_\Pi(X)$. Thus, we conclude that $\mathcal{G}_\Pi(X) = \mathcal{A}_\Pi(X)$.

"◀= " : Let $\mathcal{G}_\Pi(X) = \mathcal{A}_\Pi(X)$. Then, by the monotonicity of Cn, we have $Cn(\mathcal{G}_\Pi(X)) = Cn(\mathcal{A}_\Pi(X))$. Since X is a stable answer set of Π, by Corollary 4.3, we have $X = Cn(\mathcal{A}_\Pi(X))$. Then, by Theorem A.1, we have that X is an answer set of Π.

Proof 5.3 Let Π be a logic program and X be a stable answer set of Π.

" =▶ " : Let X be an answer set of Π. Then, by Lemma A.3, we have that X is a model of Π.

" ◀= " : Let X be a model of Π . As X is a stable answer set of , by Corollary 4.3, we have $X = Cn(\mathcal{A}_\Pi(X))$. Then, by Theorem A.1, we have that X is an answer set of Π.

Proof 6.1 Let Π be a logic program. Let X be a ι-answer set of Π and $body^-(\mathcal{A}_\Pi(X)) \cap Cn(\Pi^X) = \emptyset$. We have to show that X is a stable answer set of . Since X be a ι-answer set of , Then, we have $X = Cn(\mathcal{A}_\Pi(X))$., where $\mathcal{A}_\Pi(X)$ is a maximal subset of Π such that $body^+(\mathcal{A}_\Pi(X)) \subseteq Cn(\mathcal{A}_\Pi(X))$ and $body^-(\mathcal{A}_\Pi(X)) \cap Cn(\Pi^X) = \emptyset$. As $body^-(\mathcal{A}_\Pi(X)) \cap Cn(\Pi^X) = \emptyset$, we have that $\mathcal{A}_\Pi(X)$ satisfies Conditions (1)-(3) in Definition 3.1. Thus, X is a stable answer set of Π.

Proof 6.2 Let Π be a logic program.

Let X be a stable answer set of Π such that for every $r \in \Pi \backslash \mathcal{A}_\Pi(X)$, either

1. $body^+(r) \not\subseteq X$,

2. $body^-(r) \cap X \neq \emptyset$ *or*

3. $head(r) \in body^-(\mathcal{A}_\Pi(X) \cup \{r\})$.

We have to show that X is a ι-answer set of Π.

As X is a stable answer set of Π , by Corollary 4.3, we have $X = \mathcal{A}_\Pi(X)$. Furthermore, for every $r \in \Pi \backslash \mathcal{A}_\Pi(X)$, we have that either Condition (1),(2) or (3) holds. Then, we have that X is a ι-answer set of Π.

Lemma A.4 Let Π be a logic program and X be an alternating fixpoint of Π .

Volume : 3 | Issue : 8 | Aug 2013 | ISSN - 2249-555X

Furthermore, let $\Pi' = \{r \in \Pi \mid body^+(r) \subseteq X, body^-(r) \cap Cn(\Pi^X) = \emptyset\}$. Then, $Cn(\Pi') = X$.

Proof A.4 Since X is an alternating fixpoint of Π, we have $X = Cn(\Pi^{Cn(\Pi^X)})$. Now, we show that $Cn(\Pi') = X$.

" $\subseteq$ " : For each $r \in \Pi'$, we have $body^-(r) \cap Cn(\Pi^X) = \emptyset$. Then $body^-(\Pi) \cap Cn(\Pi^X) = \emptyset$. According to Theorem A.1, we have $Cn(\Pi') \subseteq Cn(\Pi^{Cn(\Pi^X)})$. By $X = Cn(\Pi^{Cn(\Pi^X)})$, we have $Cn(\Pi') \subseteq X$.

"$\supseteq$ ": Let $r \in \Pi$ and $r^+ \in \Pi^{Cn(\Pi^X)}$ such that $body^+(r) \subseteq Cn(\Pi')$. Then, we have either $r \in \Pi'$ or $r \notin \Pi'$. If $r \notin \Pi'$, then we have two cases: $body^+(r) \not\subseteq X$, or $body^-(r) \cap Cn(\Pi^X) \neq \emptyset$. In case $body^+(r) \not\subseteq X$, we have $r^+ \notin \Pi^{Cn(\Pi^X)}$ since $Cn(\Pi') \subseteq X$ e. This leads us to a contradiction with the assumption that $body^+(r) \subseteq Cn(\Pi')$. In case $body^-(r) \cap Cn(\Pi^X) \neq \emptyset$, we have $r^+ \notin \Pi^{Cn(\Pi^X)}$, which is a contradiction to $r^+ \in \Pi^{Cn(\Pi^X)}$. There for, let $r \in \Pi'$. Then, $head(r) \in Cn(\Pi')$ since $Cn\left(\Pi^{Cn(\Pi^X)}\right)$ is the smallest set being closed under $\Pi^{Cn(\Pi^X)}$, we have that $Cn\left(\Pi^{Cn(\Pi^X)}\right) \subseteq Cn(\Pi')$. By $X = Cn\left(\Pi^{Cn(\Pi^X)}\right)$, we have $X \subseteq Cn(\Pi')$.

Proof 7.1 Let Π be a logic program and p be an atom.

" $\Rightarrow$ " Let Π' be a stable proof of p from Π. Then, by Definition7.1, we have that

1. $p \in head(\Pi')$,
2. $body^+(\Pi') \subseteq Cn(\Pi')$,
3. $body^-(\Pi') \cap Cn(\Pi') = \emptyset$ *and*
4. $body^-(\Pi') \cap Cn\left(\Pi^{head(\Pi')}\right) = \emptyset$.

We have to show that $p \in X$ for some stable answer set X of Π. Since $Cn(\Pi')$ is closed under Π' and by (2) and (3), we have that $head(\Pi') \subseteq Cn(\Pi')$. Let $X' = Cn(\Pi')$, then by $head(\Pi') \subseteq Cn(\Pi')$, we have $head(\Pi') \subseteq X'$. We conclude that $\Pi^{X'} \subseteq \Pi^{head(\Pi')}$. The monotonicity of Cn implies that $Cn(\Pi^{X'}) \subseteq Cn(\Pi^{head(\Pi')})$. From this and (4), we have. $body^-(\Pi') \cap Cn\left(\Pi^{X'}\right) = \emptyset$. That is, $\Pi' \subseteq \Pi$ satisfies Conditions (1)-(3) in Definition 3.1. According to Corollary 4.6, there exists a stable answer set X of Π such that $\Pi' \subseteq \mathcal{A}_\Pi(X)$. Then, by the monotonicity of Cn, we have $Cn(\Pi') \subseteq Cn(\mathcal{A}_\Pi(X))$. As X is a stable answer set of Π, by corollary 4.3, we have $X = Cn(\mathcal{A}_\Pi(X))$. Then, we have $Cn(\Pi') \subseteq X$. By $X' = Cn(\Pi')$, we have $X' \subseteq X$. Clearly, we have $p \in X'$ and $p \in X$.

" $\Leftarrow$" : Let X be a stable answer set of Π such that $p \in X$.

Since, X is a stable answer set of Π, then by Definition3.1, we have that $X = Cn(\Pi')$ for some $\subseteq$-maximal $\Pi' \subseteq \Pi$, such that

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

We have to show that Π' has a stable proof of p from Π. From Conditions (1) and (2), we have that $head(\Pi') = Cn(\Pi')$. Since $X = Cn(\Pi')$ and $p \in X$. Then, $p \in \Pi'$. From this, we have that $p \in head(\Pi')$. From Condition (3) and $X = Cn(\Pi')$ and $head(\Pi') = Cn(\Pi')$, we have that, $body^-(\Pi') \cap Cn\left(\Pi^{head(\Pi')}\right) = \emptyset$. From shown above, and Definition7.1, p has a stable proof from Π, if $p \in X$ where X is a stable answer set of Π.

Proof 8.1 Let Π be a logic program, f be a selection function, and p be an atom.

"$\Leftarrow$ ": Let Π be a successful stable derivation of $\leftarrow$p from Π in context ø via f. Then, by Definition8.3, Σ consists of a finite sequence $\langle(g_0, C_0) = (\leftarrow p, \emptyset), \ldots, (g_n, C_n) = (\leftarrow \square, C_\square)\rangle$ of contextual goals and a finite sequence of rules $\langle r_1, \ldots, r_n\rangle$ such that each (g_{i+1}, C_{i+1}) is derived from (g_i, C_i) and $r_{i+1} \in \Pi$ via f, where $0 \leq i < n$. By Definition8.2, we conclude that $\bigcup_{1\leq i\leq n} head(r_i) \cap \bigcup_{1\leq i\leq n} body^-(r_i) = \emptyset$ and $\bigcup_{0\leq i\leq n} body(g_i) = \bigcup_{1\leq i\leq n} body^+(r_i) \cup \{p\}$. Since Σ is a successful stable derivation of $\leftarrow p$ from Π in context $\emptyset$, we have $\bigcup_{0\leq i\leq n} body(g_i) = \bigcup_{1\leq i\leq n} head(r_i)$ and $\bigcup_{1\leq i\leq n} body^-(r_i) \cap Cn(\Pi^{\bigcup_{0\leq i\leq n} body(g_i)}) = \emptyset$. From this and $\bigcup_{0\leq i\leq n} body(g_i) = \bigcup_{1\leq i\leq n} body^+(r_i) \cup \{p\}$, we have $\bigcup_{1\leq i\leq n} haed(r_i) = \bigcup_{1\leq i\leq n} body^+(r_i) \cup \{p\}$.

Volume : 3 | Issue : 8 | Aug 2013 | ISSN - 2249-555X

Thus, we conclude that $\bigcup_{1\leq i\leq n} body^{+}(r_i) \subseteq \bigcup_{1\leq i\leq n} haed(r_i)$. Let $\Pi' = \{r_1, \ldots, r_n\}$ be the set of rules that are used in the derivation Σ. As shown above, we have $p \in head(\Pi')$, $body^{+}(\Pi') \subseteq head(\Pi')$, $body^{-}(\Pi') \cap head(\Pi') = \emptyset$ and $body^{-}(\Pi') \cap Cn\left(\Pi^{head(\Pi')}\right) = \emptyset$. Since $(g_n, C_n) = (\leftarrow \square, C_{\square})$ it is clear that $head(\Pi') = Cn(\Pi')$. In view of Definition 7.1, we have that Π' is a stable proof for p from Π.

" =▶ " : Let Π' be a stable proof for p from Π . Then, by Theorem 7.1 $p \in X$ for some stable answer set X of Π. We have to show that there is a successful stable derivation of p from Π in context $\emptyset$.

Suppose that $\leftarrow p$ has no successful stable derivation from Π in context $\emptyset$. Then, we have that at least one Condition in Definition8.2 not satisfied.

Since X is a stable answer set of Π, by Definition3.1 , we have that $X = Cn(\Pi')$ for some $\subseteq$-maximal $\Pi' \subseteq \Pi$, such that $body^{+}(\Pi') \subseteq Cn(\Pi')$, $body^{-}(\Pi') \cap Cn(\Pi') = \emptyset, and \quad body^{-}(\Pi') \cap Cn(\Pi^{X}) = \emptyset$. Then, by Definition7.1, there exists a stable proof from Π for each atom $p \in X$. By Theorem 7.1, each $p \in X$ has a successful stable derivation from Π in context $\emptyset$. This is shows that all conditions in Definition 8.2 are satisfied. This is a contradiction for p has no successful stable derivation from Π in context $\emptyset$. So, we conclude that p has a successful stable derivation from Π in context $\emptyset$ via f.

REFERENCE [1] C. Baral. Knowledge representation, reasoning and declarative problem solving. Cambridge University Press, 2003. | [2] C. Cumbo, W. Faber, and G. Greco. Improving query optimization for disjunctive datalog. In F. Buccafurri, editor, Proceedings of the Joint Conference on Declarative Programming (AGP'03), pages 252–262, 2003. | [3] J. Delgrande, M. Gharib, R. Mercer, V. Risch, and T. Schaub. Lukaszewiczstyle answer set programming: A preliminary report. In M. De Vos and A. Provetti, editors, Proceedings of the Second International Workshop on Answer Set Programming (ASP'03), volume 78 of CEUR Workshop Proceedings, 2003. | [4] J. Dix, U. Furbach, and I. Niemel¨a. Nonmonotonic reasoning: Towards efficient calculi and implementations. In A. Voronkov and A. Robinson, editors, Handbook of Automated Reasoning, pages 1241–1354. Elselvier and MIT Press, 2001. | [5] J.Dix. A framework for representing and characterizing semantics of logic programs. In B. Nebel, C. Rich, and W. Swartout, editors, Proceedings of the Third International Conference on the Principles of Knowledge Representation and Reasoning, pages 591–602. Morgan Kaufmann Publishers, 1992. | [6] M. Gebser, M. Gharib, R. Mercer, and T. Schaub. Monotonic answer set programming. Journal of Logic and Computation, 19(4):539–564, 2009. | [7] M. Gebser, M. Gharib, and T. Schaub. Incremental answer sets and their computation. In S. Costantini and R. Watson, editors, Proceedings of the Fourth International Workshop on Answer Set Programming (ASP'07), pages 161–174, 2007. | [8] M. Gelfond and V. Lifschitz. Logic programs with classical negation. In Proceedings of the International Conference on Logic Programming, pages 579–597, 1990. | [9] M. Gharib and Amr I. Hassan. Justified answer set programming. IOSR Journal of Computer Engineering (IOSRJCE), 7(3):41–50, 2012. | [10] M. Gharib, R. Mercer, and T. Schaub. Incremental answer set programming: A preliminary report. In S. Costantini and R. Watson, editors, Proceedings of the Fourth International Workshop on Answer Set Programming (ASP'07), pages 149–160, 2007. | [11] K. Konczak, T. Linke, and T. Schaub. Graphs and colorings for answer set programming. In Computing Research Repository (CoRR), 2005. Available at http://arxiv.org/abs/cs.AI/0502082; extended version of [12]. | [12] K. Konczak, T. Linke, and T. Schaub. Graphs and colorings for answer set programming. Theory and Practice of Logic Programming 6(1-2):61–106, 2006. | [13] J. Lloyd. Foundations of Logic Programming. Symbolic Computation. 2nd edition, 1987.