Restricted Four-Valued Semantics for Answer Set Programming

Chen Chen and Zuoquan Lin(✉)

Department of Information Science, School of Mathematical Sciences,
Peking University, Beijing 100871, China
skydark2@gmail.com, linzuoquan@pku.edu.cn

Abstract. In answer set programming, an extended logic program may have no answer set, or only one trivial answer set. In this paper, we propose a new stable model semantics based on the restricted four-valued logic to overcome both inconsistences and incoherences in answer set programming. Our stable models coincide with classical answer sets when reasoning on consistent and coherent logic programs, and can be solved by transformation in existing ASP solvers. We also show the connection between our stable models and the extensions of default logic.

1 Introduction

Answer set programming (ASP) is a nonmonotonic logic programming paradigm. Its stable model semantics provides solutions of logic programs by answer sets. One limitation of the stable model semantics is that some logic programs lack answer sets, which is called incoherent programs. In extended logic programs which support both negation as failure and classical negation [12], an inconsistent program have only one trivial answer set that can infer every formula. Although it seems intuitive to detect and revise contradictions in logic programs, sometimes it is not practical and may lose information during revision. Since it is not always feasible to keep every knowledge base consistent and coherent, a paraconsistent and coherent reasoning method will be helpful since it can extract meaningful conclusions from the fragile information.

Some researchers introduced paraconsistency to answer set programming [6,14,18]. Among those, Belnap's four-valued logic [3–5], which is a paraconsistent logic that does not imply everything from inconsistent information, has been used [11,17,18] due to its intuitive semantics based on a bilattice structure.

Reiter's default logic [16] is a well-known nonmonotonic logic, which has a close relationship to answer set semantics of extended logic programs [12]. The study of handling both inconsistent and incoherent information in default logic is under investigation. The restricted four-valued default logic [8], which is based on four-valued logic, has been proposed in the presence of inconsistency and incoherence. By restricting atoms that allowed to be inconsistent, this approach can reason on both inconsistent and incoherent default theories, and also hold the same result as classical default logic while reasoning on consistent and coherent

R. Booth and M.-L. Zhang (Eds.): PRICAI 2016, LNAI 9810, pp. 68–79, 2016.
DOI: 10.1007/978-3-319-42911-3_6

theories. In this paper, we will introduce the restricted four-valued stable models inspired by the similar intuition. Our main contributions are in the following:

(1) to present a new semantics of extended logic programs which is based on the restricted four-valued logic.
(2) to show that our models can handle both inconsistencies and incoherences.
(3) to prove that our models have an 1-1 correspondence with answer set semantics on consistent and coherent programs.
(4) to give a transformation for solving our models in modern ASP solvers which supports weak constraints like DLV [13].
(5) to show that our semantics can be embedded into restricted four-valued default logic.

This paper is structured as follows. In Sect. 2, we review preliminaries. In Sect. 3, we describe our restricted four-valued semantics for extended logic programs. We show how to solve our stable models by transformation in Sect. 4. In Sect. 5, we present the connection between our models and the restricted four-valued default logic. In Sect. 6, we compare our semantics with related works. Finally, we summarize in the concluding section.

2 Preliminaries

Throughout this paper, let $\mathcal{L}$ be a propositional language, and $\mathcal{A}$ the finite set of all atoms in $\mathcal{L}$. For a literal l, we denote $atom(l)$ as the atom of l, and denote $\neg l$ as $\neg a$ if $l = a$ is an atom or a if $l = \neg a$ is a negation of an atom.

2.1 Extended Logic Programs

In [12], Gelfond and Lifschitz extend logic programming by introducing classical negation. An *extended logic program* is constructed by rules. Each rule r has the form of

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

where all l_i are literals and $n \geq m \geq 0$. The negations in literals are classical negations and the nots in rules are default negations. We denote $H(r) = \{l_0\}$ as the *head* of r, $B^+(r) = \{l_1, \cdots, l_m\}$ as the *positive body* of r, and $B^-(r) = \{l_{m+1}, \cdots, l_n\}$ as the *negative body* of r. A logic program is positive if all its rules have empty negative bodies.

An *interpretation* I, which is a literal set, satisfies a rule r, denoted as $I \models r$, iff $I \cap H(r) \neq \emptyset$ if $B^+(r) \subseteq I$ and $B^-(r) \cap I = \emptyset$. We say that an interpretation I is a *model* of a logic program P if I satisfies every rule of P. We say that I is *minimal* if there is no other model J of P that $J \subsetneq I$.

For each extended logic program P, the *Gelfond-Lifschitz reduct* of P w.r.t. an interpretation I, denoted as P^I, is an extended logic program obtained from P by deleting:

(1) each rule r that $B^-(r) \cap I$ is not an empty set;
(2) all other not l in remaining rules.

An interpretation I is an *answer set* of P, if I is the minimal model of P^I.

A logic program may have no answer set or only one trivial answer set.

Example 1. Consider the following logic programs:

(1) $P_1 = \{a \leftarrow, \neg a \leftarrow\}$;
(2) $P_2 = \{a \leftarrow \text{not } b, \neg a \leftarrow \text{not } b\}$;
(3) $P_3 = \{c \leftarrow b, \text{not } c; b \leftarrow \text{not } a\}$.

P_1 has only one trivial answer set, P_2 and P_3 have no answer set.

The extended logic programs can be embedded into Reiter's default logic [16]. A *default* d is an inference rule of form $d = \frac{\alpha:\beta_1,\ldots,\beta_n}{\gamma}$, where $\alpha, \beta_1, \ldots, \beta_n, \gamma$ are all propositional formulas. We define $Pre(d) = \alpha$ as *prerequisite* of d, $Just(d) = \{\beta_1, \ldots, \beta_n\}$ as *justification* of d, and $Con(d) = \gamma$ as *consequence* of d. A *default theory* is a pair $T = (D, W)$, where D is a set of defaults and W is a set of formulas.

An *extension* of a default theory is defined as follows.

Definition 1 ([16]). *Let $T = (D, W)$ be a default theory. For any set of formulas E, let $\Gamma(E)$ be the smallest set of formulas such that*

(1) $W \subseteq \Gamma(E)$;
(2) $Th(\Gamma(E)) = \Gamma(E)$;
(3) For any $d \in D$, if $\Gamma(E) \models_2 Pre(d)$ and $\neg\beta \notin E$ for all $\beta \in Just(d)$, then $\Gamma(E) \models_2 Con(d)$, where $\models_2$ is the classical 2-valued propositional consequence relation.

A set of formulas E is an (default) extension *of T iff $\Gamma(E) = E$, i.e. E is a fixed point of the operator Γ.*

Proposition 1 ([12]). *We identify a rule $r = l_0 \leftarrow l_1 \cdots, l_m, \text{not } l_m, \cdots, \text{not } l_n$ with the default $d(r) = \frac{l_1,\cdots,l_m:\neg l_{m+1},\cdots,\neg l_n}{l_0}$. For any extended program P, I is an answer set of P iff the deductive closure of I is an extension of $T(P)$, where $T(P) = (\{d(r) | r \in P\}, \emptyset)$ is a default theory generated by P.*

2.2 Four-Valued Logic

Belnap's four-valued logic [3–5] is based on the bilattice structure $FOUR = \{t, f, \top, \bot\}$, in which the truth values represent true, false, inconsistent and unknown respectively. The bilattice structure has two partial orderings: $\leq_t$ measures the degree of truth and $\leq_k$ measures the degree of knowledge. In this paper, we use the knowledge ordering $\leq_k$ if the using ordering is not explicitly indicated. According to the knowledge ordering, $\bot$ is the minimal element, $\top$ is the maximal element, while t and f are not comparable. Since both classical negations and default negations appear in extended logic programming, it

is necessary to distinguish explicit negative information and implicit ones. In four-valued logic, the truth value f has the explicit negative information, while $\bot$ means unknown, which has no information about whether it is true or false.

The *set of designated elements* is chosen as $\mathcal{D} = \{t, \top\}$. A *four-valued valuation* is a function that assigns a truth value from *FOUR* to each atomic formula. A valuation v *satisfies* a formula ϕ if $v(\phi) \in \mathcal{D}$. We say that v is a *model* of a set of formula S if v satisfies every formula in S. The ordering between valuations can be defined by the ordering of value on each atom, i.e. let u and v be valuations, $u \leq v$ if for each atom a we have $u(a) \leq v(a)$. A four-valued interpretation I of a logic program P can be defined by a four-valued valuation.

Definition 2 ([18]). *Let P be a positive extended program and I a four-valued interpretation, then for any rule r, $I \models r$ iff $I \models H(r)$ or there is a literal $l \in B^+(r)$ that $I \not\models l$. An interpretation I is a four-valued model of P if $I \models r$ for every rule r of P.*

A four-valued interpretation can be characterized by the literals that it satisfies, denoted as I_L. For example, we can use $I_L = \{a, \neg a, b, \neg c\}$ to denote the interpretation I which maps a to $\top$, b to t, c to f and other atoms to $\bot$. Also, let I and J be two interpretations, $I \leq J$ iff $I_L \subseteq J_L$.

3 Restricted Four-valued Semantics for Extended Logic Programs

The restricted four-valued logic is presented in [8] as the underlying logic for default reasoning. The intuition of restricting inconsistent atoms in a restricting set is a trade-off between the reasoning ability of classical logic and paraconsistency of four-valued logic.

Definition 3 ([8]). *Let S be a set of atoms. A four-valued valuation v is* restricted by *S, if $\{a \in \mathcal{A} | v(a) \notin \{t, f\}\} \subseteq S$. A four-valued valuation v is a* four-valued model of *Γ restricted by S if v is a four-valued model of Γ and restricted by S.*

Let Γ, Σ be sets of formulas. $\Gamma \models_S \Sigma$ if every four-valued model of Γ restricted by S is a four-valued model of Σ.

Based on the restricted four-valued logic, we define the *restricted reduct* as follows.

Definition 4. *Let P be an extended logic program, I an interpretation, and S a restricting set of atoms. We construct the extended logic program P_S^I obtained from P by deleting:*

(1) all not *l for each literal l that $atom(l) \in S$, and*
(2) each rule r that $l \in B^-(r)$ with any $l \in I$, and
(3) all other not *l in remaining rules.*

By the restricted reduct, we define the restricted four-valued stable models of extended logic programs in a way similar to the stable models of answer sets.

Definition 5. *Let P be an extended logic program, I an interpretation, and S a restricting set of atoms. I is a restricted four-valued stable model of P restricted by S, if I is the minimal model of P_S^I, and is restricted by S.*

I is a preferred restricted four-valued stable model of P, if I is a restricted four-valued stable model of P restricted by some restricting set S, and there is no restricted four-valued stable model of P restricted by R and R has less atoms than S by cardinality.

Notice that an interpretation I is restricted by S means that $\{a, \neg a\} \in I$ iff $a \in S$ for each atom a. The reason to compare restricting sets by cardinality is easier to calculate by existing ASP solvers as we shall see in later.

It is not surprising that the existence of restricted four-valued stable models can not be guaranteed if we set a fixed restricting set S. However, it is different if we consider all possible restricting sets.

Example 2. (Continuation of Example 1) All the logic programs in Example 1 have nontrivial and intuitive (preferred) restricted four-valued answer sets.

(1) The only preferred restricted four-valued stable model of P_1 is $\{a, \neg a\}$ restricted by $\{a\}$. This model is not trivial due to the paraconsistent underlying logic.
(2) The only preferred restricted four-valued stable model of P_2 is $\{a, \neg a\}$ restricted by $\{a\}$. It is confused that P_1 has only trivial answer set but P_2 has no answer set though they are very similar. But, P_1 and P_2 share the same preferred restricted four-valued stable model.
(3) The only preferred restricted four-valued stable model of P_3 is $\{b, c\}$ restricted by $\{c\}$. In this model, we accept b as the result of applying the second rule of P_3. The head of the first rule c is also accepted but annotated as problematic since it is included in the restricting set.

In fact, we have the following theorem to guarantee the existence of restricted four-valued stable models.

Theorem 1. *Every extended logic program P has a restricted four-valued stable model. As a result, P also has a preferred restricted four-valued stable model.*

Proof. By choosing the full atom set $\mathcal{A}$ as the restricting set, we get a positive logic program $P_I^{\mathcal{A}}$ which is independent of I. The minimal model I' of $P_I^{\mathcal{A}}$ is a restricted four-valued stable model of P restricted by $\mathcal{A}$. □

The above theorem reveals that our semantics is paraconsistent and coherent, which always has nontrivial models for every program. It is also a good news that we can keep classical stable models if they are meaningful.

Theorem 2. *Let P be an extended logic program. P has an answer set I iff I is a restricted four-valued stable model of P restricted by $\emptyset$.*

Proof. Notice that when the restricting set is an empty set, the definitions of restricted reduct and restricted four-valued stable models are all reduced to their counterparts of general answer set semantics. □

Corollary 1. *Let P be an extended logic program which has answer sets. An interpretation I is an answer set of P iff I is a preferred restricted four-valued stable model of P.*

Proof. It is easy to see by Theorem 2 and the fact that every stable model of P restricted by an empty set is preferred and only preferred. □

4 Reduction to Classical Programs

In this section we discuss how to solve (preferred) restricted four-valued stable models by transformation.

In [12], the positive form of literals has been introduced to transform extended logic programs to general logic programs with no classical negations.

Proposition 2 ([12]). *Let l be a literal. We denote the new atom l^+ as the positive form of l, The positive form of a literal set I is defined as $I^+ = \{l^+ | l \in I\}$. Let P be an extended logic program. For each rule r of P,*

$$r = l_0 \leftarrow l_1, \cdots, l_m, \text{not } l_{m+1}, \cdots, \text{not } l_n,$$

we define

$$trans(r) = l_0^+ \leftarrow l_1^+, \cdots, l_m^+, \text{not } l_{m+1}^+, \cdots, \text{not } l_n^+.$$

We define P^+ as the program by replacing each rule r by $trans(r)$. A consistent set of literals I is an answer set of P iff I^+ is an answer set of P^+.

We define our transformation in a similar way.

Definition 6. *Let P be an extended logic program. For each rule r of P,*

$$r = l_0 \leftarrow l_1, \cdots, l_m, \text{not } l_{m+1}, \cdots, \text{not } l_n,$$

we define

$$trans_r(r) = l_0^+ \leftarrow l_1^+, \cdots, l_m^+, \text{not } \neg(\neg l_{m+1})^+, \cdots, \text{not } \neg(\neg l_n)^+.$$

For each literal l, we define $neg(l) = \neg l^+ \leftarrow (\neg l)^+$. We denote $P_S^+ = \{trans_r(r) | r \in P\} \cup \{neg(l) | atom(l) \notin S\}$ as the restricting transformation *of P restricted by S.*

By the restricting transformation, we want to transform a solving problem of restricted four-valued stable models to a solving problem of answer set, which can be solved by mature ASP solvers.

Proposition 3. *Let P be an extended logic program. An interpretation I is a restricted four-valued stable model of P restricted by S iff $comp_S(I^+)$ is a consistent answer set of P_S^+, where $comp_S(I^+) = I^+ \cup \{\neg l^+ | (\neg l)^+ \in I^+ \text{ and } atom(l) \notin S\}$.*

Proof. Let I be a consistent answer set of P_S^+. We know $\neg l^+ \notin I$ for each $atom(l) \in S$, since $\neg l^+$ does not occur in any heads of rules and I is consistent. We also know that $\neg l^+ \in I$ iff $(\neg l)^+ \in I$ for each $atom(l) \notin S$, since $\neg l^+$ can only be inferred by applying rule $neg(l)$. As a result, the function $comp_S$ is an 1-1 correspondence.

By Proposition 2 and the definition of the restricted reduct, I is an answer set of P iff I^+ is a consistent answer set of P_0^+, where P_0^+ is constructed by deleting every not l in P^+ for each literal l that $atom(l) \in S$ when the restricting set S is given. So we only need to prove that $comp_S(I^+)$ is a consistent answer set of P_S^+ iff I^+ is a consistent answer set of P_0^+.

For each rule r of P,

(1) if $trans(r)$ is applied in the reduct $(P_0^+)^{I^+}$, then $B^+(trans(r)) \subseteq I^+$, and $B^-(trans(r)) \cap I^+ = \emptyset$, also $H(trans(r)) \subseteq I^+$. Since $\neg(\neg l)^+ \in comp_S(I^+)$ iff $l^+ \in I^+$, together with $B^-(trans(r)) \cap I^+ = \emptyset$, we have $B^-(trans_r(r)) \cap comp_S(I^+) = \emptyset$. Combined with that $B^+(trans_r(r)) = B^+(trans(r)) \subseteq I^+ \subseteq comp_S(I^+)$, we know that the rule $trans_r(r)$ is also appliable in the reduct $(P_S^+)^{comp_S(I^+)}$. Also $H(trans_r(r)) = H(trans(r)) \subseteq I^+ \subseteq comp_S(I^+)$;

(2) if $trans(r)$ is not applied in the reduct $(P_0^+)^{I^+}$, then
   (a) $B^+(trans(r)) \not\subseteq I^+$. We have $B^+(trans_r(r)) \not\subseteq comp_S(I^+)$ since $B^+(trans(r)) = B^+(trans_r(r))$ and $B^+(trans_r(r))$ only includes positive forms. So $trans_r(r)$ is not appliable in the reduct $(P_S^+)^{comp_S(I^+)}$ either.
   (b) or $B^-(trans(r)) \cap I^+ \neq \emptyset$. Then there is a literal l that $l^+ \in B^-(trans(r))$ and $l^+ \in I^+ \subseteq comp_S(I^+)$. We ensure that $atom(l) \notin S$ because of the construction of P_0^+. So $B^-(trans_r(r)) \cap comp_S(I^+) \supseteq \{\neg(\neg l)^+\}$.
   As a result, $trans_r(r)$ is not appliable in the reduct $(P_S^+)^{comp_S(I^+)}$ either.

Altogether, we have proved that $trans(r)$ is appliable in the reduct $(P_0^+)^{I^+}$ iff $trans_r(r)$ is appliable in the reduct $(P_S^+)^{comp_S(I^+)}$, which share the same head of rules. Therefore, I^+ is a consistent answer set of P_0^+ implies that I^+ is exactly the positive forms included in the minimal model of $(P_S^+)^{comp_S(I^+)}$, which implies $comp_S(I^+)$ is an answer set of P_S^+.

The opposite direction can be proved by the same approach. □

To solve preferred restricted four-valued stable models, we need another technique called weak constraint [7], which is a construct that extends logic programs and has been implied by some modern answer set solvers like DLV [13].

In a logic program, a *constraint* is a rule with an empty head. The only syntactic difference between constraints and weak constraints is that the symbol ←

is replaced by $:\sim$ in the weak constraints. The semantics of weak constraints is defined as follows.

Definition 7 ([13]). *Let P be an extended logic program, An interpretation I is an (optimal) answer set of P iff*

(1) I is an answer set of P without weak constraints;
(2) $H^P(I)$ is minimal over all the answer sets of P.

The objective function $H^P(I)$ maps an answer set I to a weight number by checking weak constraints. Here, we only need the simplest function that counts the number of weak constraints r that $I \supseteq B^+(r)$ and $I \cap B^-(r) = \emptyset$.

Definition 8. *Let P be an extended logic program. We define P^p as the* preferred restricting transformation *of P, where P^p contains the following rules:*

(1) $trans_r(r)$ for each rule r of P;
(2) $neg_p(l) = \neg l^+ \leftarrow (\neg l)^+, \neg(atom(l))^p$ for each literal l;
(3) $\neg a^p \leftarrow \text{not } a^p$ and $a^p \leftarrow \text{not } \neg a^p$ for each atom $a \in \mathcal{A}$;

We define P^{pp} as the program P^p with weak constraints $:\sim a^p$ for each atom $a \in \mathcal{A}$.

Theorem 3. *Let P be an extended logic program. An interpretation I is a preferred restricted four-valued stable model of P restricted by S iff $comp_S^p(I^+)$ is a consistent answer set of P^{pp}, where $comp_S^p(I^+) = comp_S(I^+) \cup \{a^p | a \in S\} \cup \{\neg a^p | a \notin S\}$.*

Proof. Notice that $a^p \in comp_S^p(I^+)$ iff $a \in S$, so $comp_S^p(I^+)$ and $comp_S(I^+)$ have an 1-1 corresponding relation. It is very direct to verify that $comp_S(I^+)$ is a consistent answer set of P_S^+ iff $comp_S^p(I^+)$ is a consistent answer set of P^p.

According to Proposition 3, we show that I is a restricted four-valued stable model of P restricted by S iff $comp_S^p(I^+)$ is a consistent answer set of P^p. The weak constraints of a^p in P^{pp} ensure that we only accept those answer sets of P^p with the minimal cardinality of S, which coincides with the definition of preferred restricted four-valued stable models. □

5 Connection with Default Logic

In this section, we show that our models have very close relation with restricted four-valued default extensions [8], which are also based on the restricted four-valued logic. The restricted four-valued extensions can be calculated by the formula transformation approach, which also introduces new atoms like positive forms. The following theorem is a limited version that only considering literals, and using symbol $(\neg a)^+$ instead of a^- in [8].

Definition 9. *For any literal set E, let $\overline{E}_S^+ = \{l^+ | l \in E\} \cup \{l^+ \leftrightarrow \neg(\neg l)^+ | atom(l) \notin S\}$. For any default rule $d = \frac{\alpha_1,\ldots,\alpha_m:\beta_1,\ldots,\beta_n}{\gamma}$, let $\overline{d}^+ = \frac{\alpha_1^+,\ldots,\alpha_m^+:\beta_1^+,\ldots,\beta_n^+}{\gamma^+}$. Let $T = (D, W)$ be a default theory. The transformed default theory $\overline{T}_S^+$ of T restricted by S, is defined as $\overline{T}_S^+ = (\overline{D}^+, \overline{W}_S^+)$, where $\overline{D}^+ = \{\overline{d}^+ | d \in D\}$.*

Theorem 4 ([8])**.** *Let $T = (D, W)$ be a default theory. E is a restricted four-valued extension of T restricted by S, iff $\overline{E}_S^+$ is a consistent extension of $\overline{T}^+$.*

We can prove that our stable model semantics can be embedded into restricted four-valued default logic, which is a paraconsistent and coherent expansion of Reiter's default logic.

Theorem 5. *Let $Th_S(I)$ be the restricted four-valued deductive closure of I restricted by S. For any extended program P, I is a restricted four-valued stable model of P restricted by S iff $Th_S(I)$ is a restricted four-valued extension of $T(P)$ restricted by S, where $T(P)$ is the default theory generated by P defined in Proposition 1.*

Proof. It can be verified by definitions that the deductive closure of $comp_S(I^+)$ is $Th(\overline{I}_S^+)$. By Proposition 3, I is a restricted four-valued stable model of P restricted by S iff $comp_S(I^+)$ is a consistent answer set of P_S^+, iff $Th(\overline{I}_S^+)$ is a consistent extension of $T(P_S^+)$ by Proposition 1. The default theory $T(P_S^+)$ can be expanded as $(\{\frac{l_1^+,\cdots,l_m^+:(\neg l_m)^+,\cdots,(\neg l_n)^+}{l_0^+} | r = l_0 \leftarrow l_1 \cdots, l_m, \text{not } l_m, \cdots, \text{not } l_n \in P\} \cup \{\frac{(\neg l)^+}{\neg l^+} | l \text{ is a literal and } atom(l) \notin S\}, \emptyset)$.

On the other hand, by formula transformation in Theorem 4, $Th_S(I)$ is a restricted four-valued extension of $T(P)$ restricted by S, iff $Th(\overline{I}_S^+)$ is a consistent extension of $\overline{T(P)}_S^+$. The default theory $\overline{T(P)}_S^+$ can be expanded as $(\{\frac{l_1^+,\cdots,l_m^+:(\neg l_m)^+,\cdots,(\neg l_n)^+}{l_0^+} | r = l_0 \leftarrow l_1 \cdots, l_m, \text{not } l_m, \cdots, \text{not } l_n \in P\}, \{\neg l^+ \leftrightarrow (\neg l)^+ | l \text{ is a literal and } atom(l) \notin S\})$, which has the same extensions of $T(P_S^+)$. □

As a corollary, our preferred restricted four-valued stable models can be embedded into a subset of preferred restricted four-valued extensions. The different is only caused by choosing preferred models by cardinality.

Proposition 4. *For any extended program P, If I is a preferred restricted four-valued stable model of P restricted by S, then $Th_S(I)$ is a preferred restricted four-valued extension of $T(P)$ restricted by S, where $T(P)$ is a default theory generated by P defined in Proposition 1.*

Proof. By Theorem 5, if I is a preferred restricted four-valued stable model of P restricted by S, $Th_S(I)$ is a restricted four-valued extension of $T(P)$ restricted by S. If $Th_S(I)$ is not preferred, there is a restricted four-valued extension

$Th_R(J)$ of $T(P)$ restricted by R and $R \subsetneq S$. By Theorem 5, J is a restricted four-valued stable model of P restricted by R which has a smaller cardinality than I restricted by S. This fact contradicts with that I is preferred. □

6 Related Works

The research on paraconsistent logic programming is even earlier than answer set semantics [6,10]. We focus on the answer set semantics of extended logic programs, which contains both nonmonotonic negation and classical negation that cause incoherences and incoherences together.

Routley semantics for answer set [14] is a paraconsistent semantics with Nelson's logic N_9 as its underlying logic. Different from their semantics, our works are based on Belnap's four-valued logic. Moreover, we do not only focus on handling inconsistent information, and also on handling incoherent information. This also distinguishes our works with other paraconsistent answer set semantics like [1].

In [18], the authors present their paraconsistent stable model semantics for extended disjunctive programs. They show that their semantics can reason with inconsistent information. For handling incoherent programs, they also provide a semi-stable (SST) model semantics by transforming original programs to disjunctive positive programs and choosing the maximally canonical models. Although this approach has many benefits, it allows redundant models even on consistent and coherent programs. For example, the program $P = \{a, b, c, d \leftarrow \text{not } a, \text{not } b; d \leftarrow \text{not } b, \text{not } c\}$ has two semi-stable models $\{a, b, c, Kb\}$ and $\{a, b, c, Ka, Kc\}$. Despite that, we have only one preferred restricted four-valued stable model $\{a, b, c\}$ which coincides with the only answer set of P.

The semi-equilibrium (SEQ) models [9,15] aim at providing an alternative coherent semantics for logic programming, which is characterized using here-and-there models. In [2], the authors figure out that SEQ-models may not respect modular structure in the rules and solve the issue by refining SEQ-models to use splitting sets. For example, the program $P = \{c \leftarrow b, \text{not } c; b \leftarrow \text{not } a\}$ has two SEQ-models (b, bc) and $(\emptyset, a)$, obviously the second model which rejects b should not be accepted since a does not occur in any heads of rules. Despite that, the only preferred restricted four-valued stable model is $\{b, c\}$ restricted by $\{c\}$, since it is not possible to reject b without proving a first. The other difference is that we treat inconsistent and incoherent as two sides of the coin. As a result, we use the same approach to solve the two problems simultaneously, but do not use paraconsistent logic to solve inconsistences only and use preferences to solve incoherences separatingly. Also, our stable models are strongly relevant to restricted four-valued default logic.

7 Conclusion

In this paper, we present the restricted four-valued stable model semantics on extended logic programs. Our semantics is paraconsistent and coherent which

benefit from the paraconsistent underlying logic. The preferred stable models correspond to the stable models if they exist and coherent. To calculate our stable models, we show a transformation which can be solved by existing ASP solvers. We reveal the relation between our models and the restricted four-valued default extensions.

We consider that the idea of restricted semantics could be applied to more applications in logic programming.

Acknowledgments. This work is partially supported by the Advance Programs Fund of Ministry of Education of China and Natural Science Foundation of China.

References

1. Alcântara, J., Damásio, C.V., Pereira, L.M.: A declarative characterisation of disjunctive paraconsistent answer sets. In: ECAI, vol. 16, p. 951. Citeseer (2004)
2. Amendola, G., Eiter, T., Leone, N.: Modular paracoherent answersets. In: Proceedings of Logics in Artificial Intelligence - 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal, 24–26 September 2014, pp. 457–471 (2014). http://dx.doi.org/10.1007/978-3-319-11558-0_32
3. Arieli, O., Avron, A.: The value of the four values. Artif. Intell. **102**(1), 97–141 (1998)
4. Belnap, N.: How a computer should think. In: Ryle, G. (ed.) Contemporary Aspects of Philosophy. Oriel Press Ltd., London (1977)
5. Belnap Jr., N.D.: A useful four-valued logic. In: Michael Dunn, J., Epstein, G. (eds.) Modern Uses of Multiple-valued Logic, pp. 5–37. Springer, Netherlands (1977)
6. Blair, H.A., Subrahmanian, V.: Paraconsistent logic programming. Theor. Comput. Sci. **68**(2), 135–154 (1989)
7. Buccafurri, F., Leone, N., Rullo, P.: Enhancing disjunctive datalog by constraints. IEEE Trans. Knowl. Data Eng. **12**(5), 845–860 (2000)
8. Chen, C., Lin, Z.: Restricted four-valued logic for default reasoning. Knowledge Science, Engineering and Management. LNCS, vol. 9403, pp. 40–52. Springer International Publishing, Cham (2015). http://dx.doi.org/10.1007/978-3-319-25159-2_4
9. Eiter, T., Fink, M., Moura, J.: Paracoherent answer set programming. In: KR, pp. 486–496 (2010)
10. Fitting, M.: Bilattices and the semantics of logic programming. J. Logic Program. **11**(2), 91–116 (1991)
11. Fitting, M.: Fixpoint semantics for logic programming a survey. Theor. Comput. Sci. **278**(1), 25–51 (2002)
12. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Gener. Comput. **9**(3–4), 365–385 (1991)
13. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV system for knowledge representation and reasoning. ACM Trans. Comput. Logic (TOCL) **7**(3), 499–562 (2006)
14. Odintsov, S., Pearce, D.J.: Routley semantics for answer sets. In: Baral, C., Greco, G., Leone, N., Terracina, G. (eds.) LPNMR 2005. LNCS (LNAI), vol. 3662, pp. 343–355. Springer, Heidelberg (2005)

15. Pearce, D.J., Valverde, A.: Quantified equilibrium logic and foundations for answer set programs. In: Garcia de la Banda, M., Pontelli, E. (eds.) ICLP 2008. LNCS, vol. 5366, pp. 546–560. Springer, Heidelberg (2008)
16. Reiter, R.: A logic for default reasoning. Artif. Intell. **13**(1), 81–132 (1980)
17. Ruet, P., Fages, F.: Combining explicit negation and negation by failure via belnap's logic. Theor. Comput. Sci. **171**(1), 61–75 (1997)
18. Sakama, C., Inoue, K.: Paraconsistent stable semantics for extended disjunctive programs. J. Logic Comput. **5**(3), 265–285 (1995)