Splitting a CR-Prolog Program

Marcello Balduccini

Intelligent Systems Department
Eastman Kodak Company
Rochester, NY 14650-2102 USA
marcello.balduccini@gmail.com

Abstract. CR-Prolog is an extension of A-Prolog, the knowledge representation language at the core of the Answer Set Programming paradigm. CR-Prolog is based on the introduction in A-Prolog of *consistency-restoring rules* (cr-rules for short), and allows an elegant formalization of events or exceptions that are unlikely, unusual, or undesired. The flexibility of the language has been extensively demonstrated in the literature, with examples that include planning and diagnostic reasoning. In this paper we hope to provide the technical means to further stimulate the study and use of CR-Prolog, by extending to CR-Prolog the Splitting Set Theorem, one of the most useful theoretical results available for A-Prolog. The availability of the Splitting Set Theorem for CR-Prolog is expected to simplify significantly the proofs of the properties of CR-Prolog programs.

1 Introduction

In recent years, Answer Set Programming (ASP) [1,2,3], a declarative programming paradigm with roots in the research on non-monotonic logic and on the semantics of default negation of Prolog, has been shown to be a useful tool for knowledge representation and reasoning (e.g., [4,5]). The underlying language, often called A-Prolog, is expressive and has a well-understood methodology of representing defaults, causal properties of actions and fluents, various types of incompleteness, etc. Over time, several extensions of A-Prolog have been proposed, aimed at improving even further the expressive power of the language.

One of these extensions, called CR-Prolog [6], is built around the introduction of *consistency-restoring rules* (cr-rules for short). The intuitive idea behind cr-rules is that they are normally not applied, even when their body is satisfied. They are only applied if the regular program (i.e., the program consisting only of conventional A-Prolog rules) is inconsistent. The language also allows the specification of a partial preference order on cr-rules, intuitively regulating the application of cr-rules.

Among the most direct uses of cr-rules is an elegant encoding of events or exceptions that are unlikely, unusual, or undesired (and preferences can be used to formalize the relative likelihood of these events and exceptions).

The flexibility of CR-Prolog has been extensively demonstrated in the literature [6,7,8,9,10], with examples including planning and diagnostic reasoning. For example, in [6], cr-rules have been used to model exogenous actions that may occur unobserved and cause malfunctioning in a physical system. In [10], instead, CR-Prolog has been used to formalize negotiations.

E. Erdem, F. Lin, and T. Schaub (Eds.): LPNMR 2009, LNCS 5753, pp. 17–29, 2009.

To further stimulate the study and use of CR-Prolog, theoretical tools are needed that simplify the proofs of the properties of CR-Prolog programs. Arguably, one of the most important such tools for A-Prolog is the Splitting Set Theorem [11]. Our goal in this paper is to extend the Splitting Set Theorem to CR-Prolog programs.

This paper is organized as follows. In the next section, we introduce the syntax and semantics of CR-Prolog. Section 3 gives key definitions and states various lemmas as well as the main result of the paper. Section 4 discusses the importance of some conditions involved in the definition of splitting set, and gives examples of the use of the Splitting Set Theorem to split CR-Prolog programs. In Section 5 we talk about related work and draw conclusions. Finally, in Section 6, we give proofs for the main results of this paper.

2 Background

The syntax and semantics of ASP are defined as follows. Let Σ be a signature containing constant, function, and predicate symbols. Terms and atoms are formed as usual. A literal is either an atom a or its strong (also called classical or epistemic) negation $\neg a$. The *complement* of an atom a is literal $\neg a$, while the complement of $\neg a$ is a. The complement of literal l is denoted by $\overline{l}$. The sets of atoms and literals formed from Σ are denoted by $atoms(\Sigma)$ and $lit(\Sigma)$, respectively.

A *regular rule* is a statement of the form:

$$[r]\ h_1 \text{ OR } h_2 \text{ OR } \ldots \text{ OR } h_k \leftarrow l_1, \ldots, l_m, \text{not } l_{m+1}, \ldots, \text{not } l_n \qquad (1)$$

where r, called *name*, is a *possibly compound* term uniquely denoting the regular rule, h_i's and l_i's are literals and *not* is the so-called *default negation*. The intuitive meaning of the regular rule is that a reasoner who believes $\{l_1, \ldots, l_m\}$ and has no reason to believe $\{l_{m+1}, \ldots, l_n\}$, must believe one of h_i's.

A *consistency-restoring rule* (or *cr-rule*) is a statement of the form:

$$[r]\ h_1 \text{ OR } h_2 \text{ OR } \ldots \text{ OR } h_k \stackrel{+}{\leftarrow} l_1, \ldots l_m, \text{not } l_{m+1}, \ldots, \text{not } l_n \qquad (2)$$

where r, h_i's, and l_i's are as before. The intuitive reading of a cr-rule is that a reasoner who believes $\{l_1, \ldots, l_m\}$ and has no reason to believe $\{l_{m+1}, \ldots, l_n\}$, *may possibly* believe one of h_i's. The implicit assumption is that this possibility is used as little as possible, only when the reasoner cannot otherwise form a consistent set of beliefs. A preference order on the use of cr-rules is expressed by means of the atoms of the form $prefer(r_1, r_2)$. Such an atom informally says that r_2 should not be used unless there is no way to obtain a consistent set of beliefs with r_1. More details on preferences in CR-Prolog can be found in [6,12,13].

By *rule* we mean a regular rule or a cr-rule. Given a rule ρ of the form (1) or (2), we call $\{h_1, \ldots, h_k\}$ the *head* of the rule, denoted by $head(\rho)$, and $\{l_1, \ldots, l_m, \text{not } l_{m+1}, \ldots, \text{not } l_n\}$ its *body*, denoted by $body(\rho)$. Also, $pos(\rho)$ denotes $\{l_1, \ldots, l_m\}$, $neg(\rho)$ denotes $\{l_{m+1}, \ldots, l_n\}$, $name(\rho)$ denotes name r, and $lit(\rho)$ denotes the set of all literals from ρ. When $l \in lit(\rho)$, we say that l *occurs* in ρ.

A *program* is a pair $\langle \Sigma, \Pi \rangle$, where Σ is a signature and Π is a set of rules over Σ. Often we denote programs by just the second element of the pair, and let the signature be defined implicitly. In that case, the signature of Π is denoted by $\Sigma(\Pi)$.

In practice, variables are often allowed to occur in ASP programs. A rule containing variables (called a *non-ground* rule) is then viewed as a shorthand for the set of its *ground instances*, obtained by replacing the variables in it by all of the possible ground terms. Similarly, a non-ground program is viewed as a shorthand for the program consisting of the ground instances of its rules.

Given a program Π, $\mu(\Pi)$ denotes the set of names of the rules from Π. In the rest of the discussion, letter r (resp., ρ), possibly indexed, denotes the name of a rule (resp., a rule). Given a set of rule names R, $\rho(R, \Pi)$ denotes the set of rules from Π whose name is in R. $\rho(r, \Pi)$ is shorthand for $\rho(\{r\}, \Pi)$. To simplify notation, we allow writing $r \in \Pi$ to mean that a rule with name r is in Π. We extend the use of the other set operations in a similar way. Also, given a program Π, $head(r)$ and $body(r)$ denote[1] the corresponding parts of $\rho(r, \Pi)$. Given a CR-Prolog program, Π, the *regular part* of Π is the set of its regular rules, and is denoted by $reg(\Pi)$. The set of cr-rules of Π is denoted by $cr(\Pi)$. Programs that do not contain cr-rules are legal ASP programs, and their semantics is defined as usual. Next, we define the semantics of arbitrary CR-Prolog programs. Let us begin by introducing some notation.

For every $\mathcal{R} \subseteq cr(\Pi)$, $\theta(\mathcal{R})$ denotes the set of regular rules obtained from $\mathcal{R}$ by replacing every connective $\stackrel{+}{\leftarrow}$ with $\leftarrow$. Given a program Π and a set R of rule names, $\theta(R)$ denotes the application of θ to the rules of Π whose name is in R.

A literal l is *satisfied* by a set of literals S ($S \models l$) if $l \in S$. An expression not l is satisfied by S if $l \notin S$. The body of a rule is satisfied by S if each element of the set is satisfied by S. A set of literals S entails $prefer^*(r_1, r_2)$ ($S \models prefer^*(r_1, r_2)$) if:

– $S \models prefer(r_1, r_2)$, or
– $S \models prefer(r_1, r_3)$ and $S \models prefer^*(r_3, r_2)$.

The semantics of CR-Prolog is given in three steps.

Definition 1. Let S be a set of literals and R be a set of names of cr-rules from Π. The pair $\mathcal{V} = \langle S, R \rangle$ is a *view* of Π if:

1. S is an answer set[2] of $reg(\Pi) \cup \theta(R)$, and
2. for every r_1, r_2, if $S \models prefer^*(r_1, r_2)$, then $\{r_1, r_2\} \not\subseteq R$, and
3. for every r in R, $body(r)$ is satisfied by S.

We denote the elements of $\mathcal{V}$ by $\mathcal{V}^S$ and $\mathcal{V}^R$ respectively. The cr-rules in $\mathcal{V}^R$ are said to be *applied*. This definition of view differs from the one given in previous papers (e.g., [6,13]) in that set R here is a set of names of cr-rules rather than a set of cr-rules. The change allows one to simplify the proofs of the theorems given later. Because of the one-to-one correspondence between cr-rules and their names, the two definitions are equivalent.

[1] The notation can be made more precise by specifying Π as an argument, but in the present paper Π will always be clear from the context.

[2] We only consider *consistent* answer sets.

For every pair of views of Π, $\mathcal{V}_1$ and $\mathcal{V}_2$, $\mathcal{V}_1$ *dominates* $\mathcal{V}_2$ if there exist $r_1 \in \mathcal{V}_1^R$, $r_2 \in \mathcal{V}_2^R$ such that $(\mathcal{V}_1^S \cap \mathcal{V}_2^S) \models prefer^*(r_1, r_2)$.

Definition 2. A view, $\mathcal{V}$, is a *candidate answer set* of Π if, for every view $\mathcal{V}'$ of Π, $\mathcal{V}'$ does not dominate $\mathcal{V}$.

Definition 3. A set of literals, S, is an *answer set* of Π if:

1. there exists a set R of names of cr-rules from Π such that $\langle S, R \rangle$ is a candidate answer set of Π, and
2. for every candidate answer set $\langle S', R' \rangle$ of Π, $R' \not\subset R$.

3 Splitting Set Theorem

Proceeding along the lines of [11], we begin by introducing the notion of splitting set for a CR-Prolog program, and then use this notion to state the main theorems.

A *preference set for cr-rule* r *with respect to a set of literals* S is the set

$$\pi(r, S) = \{r' \mid S \models prefer^*(r, r') \text{ or } S \models prefer^*(r', r)\}.$$

Given a program Π, the preference set of r with respect to the literals from the signature of Π is denoted by $\pi(r)$.

Definition 4. *Literal* l *is* relevant to cr-rule r *(for short,* l *is* r*-relevant) if:*

1. l *occurs in* r*, or*
2. l *occurs in some rule where a literal relevant to* r *occurs, or*
3. $\overline{l}$ *is relevant to* r*, or*
4. $l = prefer(r, r')$ *or* $l = prefer(r', r)$*.*

Definition 5. *A* splitting set *for a program* Π *is a set* U *of literals from* $\Sigma(\Pi)$ *such that:*

– *for every rule* $r \in \Pi$*, if* $head(r) \cap U \neq \emptyset$*, then* $lit(r) \subseteq U$*;*
– *for every cr-rule* $r \in \Pi$*, if some* $l \in U$ *is relevant to* r*, then every* r*-relevant literal belongs to* U*.*

Observation 1. *For programs that do not contain cr-rules, this definition of splitting set coincides with the one given in [11].*

Observation 2. *For every program* Π *and splitting set* U *for* Π*, if* $l \in U$ *is* r*-relevant and* $r' \in \pi(r)$*, then every* r'*-relevant literal from* $\Sigma(\Pi)$ *belongs to* U*.*

We define the notions of bottom and partial evaluation of a program similarly to [11]. The bottom of a CR-Prolog program Π relative to splitting set U is denoted by $b_U(\Pi)$ and consists of every rule $\rho \in \Pi$ such that $lit(\rho) \subseteq U$. Given a program Π and a set R of names of rules, $b_U(R)$ denotes the set of rule names in $b_U(\rho(R, \Pi))$.

The *partial evaluation* of a CR-Prolog program Π w.r.t. splitting set U and set of literals X, denoted by $e_U(\Pi, X)$, is obtained as follows:

- For every rule $\rho \in \Pi$ such that $pos(\rho) \cap U$ is part of X and $neg(\rho) \cap U$ is disjoint from X, $e_U(\Pi, X)$ contains the rule ρ' such that:

$$name(\rho') = name(\rho), \quad head(\rho') = head(\rho),$$
$$pos(\rho') = pos(\rho) \setminus U, \quad neg(\rho') = neg(\rho) \setminus U.$$

- For every other rule $\rho \in \Pi$, $e_U(\Pi, X)$ contains the rule ρ' such that:

$$name(\rho') = name(\rho), \quad head(\rho') = head(\rho),$$
$$pos(\rho') = \{\bot\} \cup pos(\rho) \setminus U, \quad neg(\rho') = neg(\rho) \setminus U.$$

Given Π, U, and X as above, and a set R of names of rules from Π, $e_U(R, X)$ denotes the set of rule names in $e_U(\rho(R, \Pi), X)$.

Observation 3. *For every program Π, splitting set U and set of literals X, $e_U(\Pi, X)$ is equivalent to the similarly denoted set of rules defined in [11].*

Observation 4. *For every program Π, set R of names of rules from Π, splitting set U, and set of literals X, $R = e_U(R, X)$.*

From Observations 1 and 3, and from the original Splitting Set Theorem [11], one can easily prove the following statement.

Theorem 1 (Splitting Set Theorem from [11]). *Let U be a splitting set for a program Π that does not contain any cr-rule. A set S of literals is an answer set of Π if and only if: (i) X is an answer set of $b_U(\Pi)$; (ii) Y is an answer set of $e_U(\Pi \setminus b_U(\Pi), X)$; (iii) $S = X \cup Y$ is consistent.*

We are now ready to state the main results of this paper. Complete proofs can be found in Section 6.

Lemma 1 (Splitting Set Lemma for Views). *Let U be a splitting set for a program Π, S a set of literals, and R a set of names of cr-rules from Π. The pair $\langle S, R\rangle$ is a view of Π if and only if:*

- *$\langle X, b_U(R)\rangle$ is a view of $b_U(\Pi)$;*
- *$\langle Y, R \setminus b_U(R)\rangle$ is a view of $e_U(\Pi \setminus b_U(\Pi), X)$;*
- *$S = X \cup Y$ is consistent.*

Lemma 2 (Splitting Set Lemma for Candidate Answer Sets). *Let U be a splitting set for a program Π. A pair $\langle S, R\rangle$ is a candidate answer set of Π if and only if:*

- *$\langle X, b_U(R)\rangle$ is a candidate answer set of $b_U(\Pi)$;*
- *$\langle Y, R \setminus b_U(R)\rangle$ is a candidate answer set of $e_U(\Pi \setminus b_U(\Pi), X)$;*
- *$S = X \cup Y$ is consistent.*

Theorem 2 (Splitting Set Theorem for CR-Prolog). *Let U be a splitting set for a program Π. A consistent set of literals S is an answer set of Π if and only if:*

- *X is an answer set of $b_U(\Pi)$;*
- *Y is an answer set of $e_U(\Pi \setminus b_U(\Pi), X)$;*
- *$S = X \cup Y$ is consistent.*

4 Discussion

Now that the Splitting Set Theorem for CR-Prolog has been stated, in this section we give examples of the application of the theorem and discuss the importance of the conditions of Definition 4 upon which the definition of splitting set and the corresponding theorem rely.

Let us begin by examining the role of the conditions of Definition 4.

Consider condition (3) of Definition 4. To see why the condition is needed, consider the following program, P_1 (as usual, rule names are omitted whenever possible to simplify the notation):

$$[r_1]\ q \stackrel{+}{\leftarrow} \text{not } p.$$

$$s \leftarrow \text{not } q.$$

$$\neg s.$$

It is not difficult to see that P_1 has the unique answer set $\{q, \neg s\}$, intuitively obtained from the application of r_1. Let us now consider set $U_1 = \{q, p, s\}$. Notice that U_1 satisfies the definition of splitting set, as long as the condition under discussion is dropped from Definition 4. The corresponding $b_{U_1}(P_1)$ is:

$$[r_1]\ q \stackrel{+}{\leftarrow} \text{not } p.$$

$$s \leftarrow \text{not } q.$$

$b_{U_1}(P_1)$ has a unique answer set, $X_1^a = \{s\}$, obtained without applying r_1. $e_{U_1}(P_1 \setminus b_{U_1}(P_1), \{s\})$ is:

$$\neg s.$$

which has a unique answer set, $Y_1^a = \{\neg s\}$. Notice that $X_1^a \cup Y_1^a$ is inconsistent. Because X_1^a and Y_1^a are unique answer sets of the corresponding programs, it follows that the answer set of P_1 cannot be obtained from the any of the answer sets of $b_{U_1}(P_1)$ and of the corresponding partial evaluation of P_1. Hence, dropping condition (3) of Definition 4 causes the splitting set theorem to no longer hold.

Very similar reasoning shows the importance of condition (2): just obtain P_2 from P_1 by (i) replacing $\neg s$ in P_1 by t and (ii) adding a constraint $\leftarrow t, s$, and consider the set $U_2 = \{q, p\}$. Observe that, if the condition is dropped, then U_2 is a splitting set for P_2, but the splitting set theorem does not hold.

Let us now focus on condition (4). Consider program P_3:

$$[r_1]\ q \stackrel{+}{\leftarrow} \text{not } p.$$
$$[r_2]\ s \stackrel{+}{\leftarrow} \text{not } t.$$

$$prefer(r_2, r_1).$$

$$\leftarrow \text{not } q.$$
$$\leftarrow \text{not } s.$$

Observe that P_3 is inconsistent, the intuitive explanation being that r_1 can only be used if there is no way to use r_2 to form a consistent set of beliefs, but the only way to form such a consistent set would be to use r_1 and r_2 together. Now consider set $U_3 = \{q, p, prefer(r_2, r_1)\}$. The corresponding $b_{U_3}(P_3)$ is:

$$[r_1]\ q \stackrel{+}{\leftarrow} \text{not } p.$$

$$prefer(r_2, r_1).$$

$$\leftarrow \text{not } q.$$

which has a unique answer set $X_3 = \{q, prefer(r_2, r_1)\}$. The partial evaluation $e_{U_3}(P_3 \setminus b_{U_3}(P_3), X_3)$ is:

$$[r_2]\ s \stackrel{+}{\leftarrow} \text{not } t.$$
$$\leftarrow \text{not } s.$$

whose unique answer set is $Y_3 = \{s\}$. If condition (4) is dropped from Definition 4, then U_3 is a splitting set for P_3. However, the splitting set theorem does not hold, as P_3 is inconsistent while $X_3 \cup Y_3$ is consistent.

Let us now give a few examples of the use of the Splitting Set Theorem to finding the answer sets of CR-Prolog programs. Consider program P_4:

$$[r_1]\ q \stackrel{+}{\leftarrow} \text{not } a.$$
$$[r_2]\ p \stackrel{+}{\leftarrow} \text{not } t.$$
$$a \text{ OR } b.$$
$$s \leftarrow \text{not } b.$$
$$\leftarrow \text{not } q, b.$$
$$c \text{ OR } d.$$
$$u \leftarrow z, \text{not } p.$$
$$z \leftarrow \text{not } u.$$

and the set $U_4 = \{q, a, b, s\}$. It is not difficult to check the conditions and verify that U_4 is a splitting set for P_4. In particular, observe that U_4 includes the r_1-relevant literals, and does not include the r_2-relevant literals. $b_{U_4}(P_4)$ is:

$$[r_1]\ q \stackrel{+}{\leftarrow} \text{not } a.$$
$$a \text{ OR } b.$$
$$s \leftarrow \text{not } b.$$
$$\leftarrow \text{not } q, b.$$

Because P_4 contains a single cr-rule, from the semantics of CR-Prolog it follows that its answer sets are those of $reg(b_{U_4}(P_4))$, if the program is consistent, and those of $reg(b_{U_4}(P_4)) \cup \theta(\{r_1\})$ otherwise. $reg(b_{U_4}(P_4))$ has a unique answer set, $X_4 = \{a, s\}$, which is, then, also the answer set of $b_{U_4}(P_4)$. The partial evaluation $e_{U_4}(P_4 \setminus b_{U_4}(P_4), X_4)$ is:

$$[r_2]\ p \stackrel{+}{\leftarrow} \text{not } t.$$
$$c \text{ OR } d.$$
$$u \leftarrow z, \text{not } p.$$
$$z \leftarrow \text{not } u.$$

Again, the program contains a single cr-rule. This time, $reg(e_{U_4}(P_4 \setminus b_{U_4}(P_4), X_4))$ is inconsistent. $reg(e_{U_4}(P_4 \setminus b_{U_4}(P_4), X_4)) \cup \theta(\{r_2\})$, on the other hand, has an answer set $Y_4 = \{p, c, z\}$, which is, then, also an answer set of $e_{U_4}(P_4 \setminus b_{U_4}(P_4), X_4)$. Therefore, an answer set of P_4 is

$$X_4 \cup Y_4 = \{a, s, p, c, z\}.$$

Now the following modification of P_4, P_5:

$$\begin{array}{l}
[r_1]\ q \stackrel{+}{\leftarrow} \text{not } a. \\
[r_2]\ p \stackrel{+}{\leftarrow} \text{not } t. \\
a \text{ OR } b. \\
s \leftarrow \text{not } b. \\
\leftarrow \text{not } q, b. \\
c \text{ OR } d \leftarrow v. \\
\neg c \leftarrow \text{not } v. \\
u \leftarrow z, \text{not } p. \\
z \leftarrow \text{not } u. \\
v \leftarrow \text{not } w.
\end{array}$$

The goal of this modification is to show how rules, whose literals are not relevant to any cr-rule, can be split. Let U_5 be $\{q, a, b, s, v, w\}$. Notice that U_5 is a splitting set for P_5 even though $v \in U_5$ and P_5 contains the rule c OR $d \leftarrow v$. In fact, v is not relevant to any cr-rule from P_5, and thus c and d are not required to belong to U_5. $b_{U_5}(P_5)$ is:

$$\begin{array}{l}
[r_1]\ q \stackrel{+}{\leftarrow} \text{not } a. \\
a \text{ OR } b. \\
s \leftarrow \text{not } b. \\
\leftarrow \text{not } q, b. \\
v \leftarrow \text{not } w.
\end{array}$$

which has an answer set $X_5 = \{a, s, v\}$. $e_{U_5}(P_5 \setminus b_{U_5}(P_5), X_5)$ is:

$$\begin{array}{l}
[r_2]\ p \stackrel{+}{\leftarrow} \text{not } t. \\
c \text{ OR } d. \\
\neg c \leftarrow \bot. \\
u \leftarrow z, \text{not } p. \\
z \leftarrow \text{not } u.
\end{array}$$

which has an answer set $Y_5 = \{p, c, z\}$. Hence, an answer set of P_5 is

$$X_5 \cup Y_5 = \{a, s, v, p, c, z\}.$$

5 Related Work and Conclusions

Several papers have addressed the notion of splitting set and stated various versions of splitting set theorems throughout the years. Notable examples are [11], with the original formulation of the Splitting Set Theorem for A-Prolog, [14], with a Splitting Set Theorem for default theories, and [15] with a Splitting Set Theorem for epistemic specifications.

In this paper we have defined a notion of splitting set for CR-Prolog programs, and stated the corresponding Splitting Set Theorem. We hope that the availability of this theoretical result will further stimulate the study and use of CR-Prolog, by making it easier to prove the properties of the programs written in this language. As the reader may have noticed, to hold for CR-Prolog programs (that include at least one cr-rule), the Splitting Set Theorem requires substantially stronger conditions than the Splitting Set Theorem for A-Prolog. We hope that future research will allow weakening the conditions of the theorem given here, but we suspect that the need for stronger conditions is strictly tied to the nature of cr-rules.

6 Proofs

Proof of Lemma 1. To be a view of a program, a pair $\langle S, R \rangle$ must satisfy all of the requirements of Definition 1. Let us begin from item (1) of the definition. We must show that S is an answer set of $reg(\Pi) \cup \theta(R)$ if and only if:

- X is an answer set of $reg(b_U(\Pi)) \cup \theta(b_U(R))$;
- Y is an answer set of $reg(e_U(\Pi \setminus b_U(\Pi), X)) \cup \theta(R \setminus b_U(R))$.

From Theorem 1, S is an answer set of $reg(\Pi) \cup \theta(R)$ iff:

- X is an answer set of

$$\begin{aligned} &b_U(reg(\Pi) \cup \theta(R)) = \\ &b_U(reg(\Pi)) \cup b_U(\theta(R)) = \\ &b_U(reg(\Pi)) \cup \theta(b_U(R)) = \\ &reg(b_U(\Pi)) \cup \theta(b_U(R)). \end{aligned}$$

- Y is an answer set of

$$\begin{aligned} &e_U((reg(\Pi) \cup \theta(R)) \setminus b_U(reg(\Pi) \cup \theta(R)), X) = \\ &e_U((reg(\Pi) \cup \theta(R)) \setminus (reg(b_U(\Pi)) \cup \theta(b_U(R))), X) = \\ &e_U((reg(\Pi) \setminus reg(b_U(\Pi))) \cup (\theta(R) \setminus \theta(b_U(R))), X) = \\ &e_U(reg(\Pi \setminus b_U(\Pi)) \cup \theta(R \setminus b_U(R)), X) = \\ &e_U(reg(\Pi \setminus b_U(\Pi)), X) \cup e_U(\theta(R \setminus b_U(R)), X) = \\ &reg(e_U(\Pi \setminus b_U(\Pi), X)) \cup \theta(e_U(R \setminus b_U(R), X)) = \\ &reg(e_U(\Pi \setminus b_U(\Pi), X)) \cup \theta(R \setminus b_U(R)), \end{aligned}$$

  where the last transformation follows from Observation 4.
- $S = X \cup Y$ is consistent.

This completes the proof for item (1) of the definition of view, and furthermore concludes that $S = X \cup Y$ is consistent. Let us now consider item (2) of the definition of view. We must show that

$$\forall r_1, r_2 \in R, \text{ if } S \models prefer^*(r_1, r_2), \text{ then } \{r_1, r_2\} \not\subseteq R$$
$$\text{iff}$$
$$\forall r_1, r_2 \in b_U(R), \text{ if } X \models prefer^*(r_1, r_2), \text{ then } \{r_1, r_2\} \not\subseteq b_U(R), \text{ and}$$
$$\forall r_1, r_2 \in R \setminus b_U(R), \text{ if } Y \models prefer^*(r_1, r_2), \text{ then } \{r_1, r_2\} \not\subseteq R \setminus b_U(R).$$

Left-to-right. The statement follows from the fact that $X \subseteq S$, $Y \subseteq S$, and $b_U(R) \subseteq R$.

Right-to-left. Proceeding by contradiction, suppose that, for some $r_1, r_2 \in R$, $S \models prefer^*(r_1, r_2)$, but $\{r_1, r_2\} \subseteq R$. By definition of preference set, $r_2 \in \pi(r_1)$. Let l be some literal from $head(r_1)$. Obviously, either $l \in U$ or $l \notin U$.

Suppose $l \in U$. By the definition of splitting set, $prefer(r_i, r_j) \in U$ for every $r_i, r_j \in \{r_1\} \cup \pi(r_1)$. Moreover, for every $r' \in \pi(r_1)$ and every $l' \in head(r')$, l' belongs to U. But $r_2 \in \pi(r_1)$. Hence, $X \models prefer^*(r_1, r_2)$ and $\{r_1, r_2\} \subseteq b_U(R)$. *Contradiction.*

Now suppose $l \notin U$. With reasoning similar to the previous case, we can conclude that $prefer(r_i, r_j) \in \overline{U}$ for every $r_i, r_j \in \{r_1\} \cup \pi(r_1)$. Moreover, for every $r' \in \pi(r_1)$ and every $l' \in head(r')$, l' belongs to $\overline{U}$. Because $r_2 \in \pi(r_1)$, $Y \models prefer^*(r_1, r_2)$ and $\{r_1, r_2\} \subseteq R \setminus b_U(R)$. *Contradiction.*

This completes the proof for item (2) of the definition of view. Let us now consider item (3). We must prove that

$$\forall r \in R, body(\rho(r, \Pi)) \text{ is satisfied by } S$$
$$\text{iff}$$
$$\forall r \in b_U(R), body(\rho(r, b_U(\Pi))) \text{ is satisfied by } X, \text{ and}$$
$$\forall r \in R \setminus b_U(R), body(\rho(r, e_U(\Pi \setminus b_U(\Pi), X))) \text{ is satisfied by } Y.$$

Left-to-right. The claim follows from the following observations: (i) for every $r \in R$, either $r \in b_U(R)$ or $r \in R \setminus b_U(R)$; (ii) $body(\rho(r, e_U(\Pi \setminus b_U(\Pi), X)))$ is satisfied by Y iff $body(\rho(r, \Pi))$ is satisfied by $X \cup Y = S$.

Right-to-left. Again, observe that, for every $r \in R$, either $r \in b_U(R)$ or $r \in R \setminus b_U(R)$.

Suppose $r \in b_U(R)$. Because $X \subseteq S$, $body(\rho(r, b_U(\Pi)))$ is satisfied by S. Because $b_U(\Pi) \subseteq \Pi$, $\rho(r, b_U(\Pi)) = \rho(r, \Pi)$.

Suppose $r \in R \setminus b_U(R)$. The notion of partial evaluation is defined in such a way that, if $body(\rho(r, e_U(\Pi \setminus b_U(\Pi), X)))$ is satisfied by Y, then $body(\rho(r, \Pi))$ is satisfied by $X \cup Y = S$.

Proof of Lemma 2. From Lemma 1, it follows that $\langle S, R\rangle$ is a view of Π iff (i) $\langle X, b_U(R)\rangle$ is a view of $b_U(\Pi)$, (ii) $\langle Y, R \setminus b_U(R)\rangle$ is a view of $e_U(\Pi \setminus b_U(\Pi), X)$, and (iii) $S = X \cup Y$ is consistent. Therefore, we only need to prove that:

$$\text{no view of } \Pi \text{ dominates } \langle S, R\rangle \tag{3}$$

if and only if

$$\text{no view of } b_U(\Pi) \text{ dominates } \langle X, b_U(R)\rangle, \text{ and} \tag{4}$$

$$\text{no view of } e_U(\Pi \setminus b_U(\Pi), X) \text{ dominates } \langle Y, R \setminus b_U(R)\rangle. \tag{5}$$

Left-to-right. Let us prove that (3) implies (4). By contradiction, suppose that:

$$\text{there exists a view } \mathcal{V}'_X = \langle X', R'_X\rangle \text{ of } b_U(\Pi) \text{ dominates } \mathcal{V}_X = \langle X, b_U(R)\rangle. \tag{6}$$

Let (X'_D, X'_I) be the partition of X' such that X'_D is the set of the literals from X' that are relevant to the cr-rules in $b_U(R) \cup R'_X$. Let (X_D, X_U) be a similar partition of X. From Definition 4, it is not difficult to see that $\langle X_I \cup X'_D, R'_X\rangle$ is a view of $b_U(\Pi)$. Moreover, given $R' = (R \setminus b_U(R)) \cup R'_X$, $\langle Y, R' \setminus b_U(R')\rangle$ is a view of $e_U(\Pi \setminus b_U(\Pi), X_I \cup X'_D)$. By Lemma 1, $\langle X' \cup Y, R'\rangle$ is a view of Π. From hypothesis (6), it follows that there exist $r \in b_U(R) \subseteq R$ and $r' \in b_U(R') \subseteq R'$ such that $(X \cap X') \models prefer^*(r', r)$. But then $\langle X' \cup Y, R'\rangle$ dominates $\langle S, R\rangle$. *Contradiction.*

Let us prove that (3) implies (5). By contradiction, suppose that there exists a view $\mathcal{V}'_Y = \langle Y', R'_Y\rangle$ of $e_U(\Pi \setminus b_U(\Pi), X)$ that dominates $\mathcal{V}_Y = \langle Y, R \setminus b_U(R)\rangle$. That is,

$$\text{there exist } r \in R \setminus b_U(R),\ r' \in R'_Y \text{ such that } (Y \cap Y') \models prefer^*(r', r). \tag{7}$$

Let R' be $R'_Y \cup b_U(R)$. By Lemma 1, $\langle X \cup Y', R'\rangle$ is a view of Π. From (7), it follows that there exist $r \in R$, $r' \in R'$ such that $(Y \cap Y') \models prefer^*(r', r)$. Therefore, $\langle X \cup Y', R'\rangle$ dominates $\langle S, R\rangle$. *Contradiction.*

Next, from (4) and (5), we prove (3). By contradiction, suppose that there exists a view $\mathcal{V}' = \langle S', R'\rangle$ of Π that dominates $\mathcal{V} = \langle S, R\rangle$. That is, there exist $r \in R$, $r' \in R'$ such that $(S \cap S') \models prefer^*(r', r)$. There are two cases: $head(r') \subseteq U$ and $head(r') \subseteq \overline{U}$.

Case 1: $head(r') \subseteq U$. From Lemma 1 it follows that $\langle S \cap U, b_U(R)\rangle$ is a view of $b_U(\Pi)$. Similarly, $\langle S' \cap U, b_U(R')\rangle$ is a view of $b_U(\Pi)$. Because $head(r') \subseteq U$, from the definition of splitting set it follows that: (i) $head(r) \subseteq U$; (ii) because $(S \cap S') \models prefer^*(r', r)$, $(S \cap S' \cap U) \models prefer^*(r', r)$ also holds. Therefore, $\langle S' \cap U, b_U(R')\rangle$ dominates $\langle S \cap U, b_U(R)\rangle$. *Contradiction.*

Case 2: $head(r') \subseteq \overline{U}$. From Lemma 1 it follows that $\langle S \setminus U, R \setminus b_U(R)\rangle$ is a view of $e_U(\Pi \setminus b_U(\Pi), S \cap U)$. Similarly, $\langle S' \setminus U, R' \setminus b_U(R')\rangle$ is a view of $e_U(\Pi \setminus b_U(\Pi), S' \cap U)$. Because $head(r') \subseteq \overline{U}$, from the definition of splitting set it follows that: (i) $head(r) \subseteq \overline{U}$; (ii) because $(S \cap S') \models prefer^*(r', r)$, $(S \setminus U) \cap (S' \setminus U) \models prefer^*(r', r)$ also holds.

Consider now set $Q' \subseteq S' \setminus U$, consisting of all of the literals of $S' \setminus U$ that are relevant to the cr-rules of $R \cup R' \setminus b_U(R \cup R')$. Also, let $Q \subseteq S \setminus U$ be the set of all of the literals of $S \setminus U$ that are relevant to the cr-rules of $R \cup R' \setminus b_U(R \cup R')$, and $\overline{Q} = S \setminus U \setminus Q$. That is, $\overline{Q}$ is the set of literals from $S \setminus U$ that are *not* relevant to any cr-rule of $R \cup R' \setminus b_U(R \cup R')$.

From Definition 4, it is not difficult to conclude that $\langle \overline{Q} \cup Q', R' \setminus b_U(R')\rangle$ is a view of $e_U(\Pi \setminus b_U(\Pi), S \cap U)$. Furthermore, $(S \setminus U) \cap (Q \cup Q') \models prefer^*(r', r)$. Therefore, $\langle \overline{Q} \cup Q', R' \setminus b_U(R')\rangle$ dominates $\langle S \setminus U, R \setminus b_U(R)\rangle$. *Contradiction.*

Proof of Theorem 2. From the definition of answer set and Lemma 2, it follows that there exists a set R of (names of) cr-rules from Π such that $\langle S, R\rangle$ is a candidate answer set of Π if and only if:

- $\langle X, b_U(R)\rangle$ is a candidate answer set of $b_U(\Pi)$;
- $\langle Y, R \setminus b_U(R)\rangle$ is a candidate answer set of $e_U(\Pi \setminus b_U(\Pi), X)$;
- $S = X \cup Y$ is consistent.

Therefore, we only need to prove that:

$$\text{for every candidate answer set } \langle S', R'\rangle \text{ of } \Pi,\ R' \not\subset R \tag{8}$$

if and only if

$$\text{for every candidate answer set } \langle S'_X, R'_X\rangle \text{ of } b_U(\Pi),\ R'_X \not\subset b_U(R), \text{ and} \tag{9}$$

$$\text{for every candidate answer set } \langle S'_Y, R'_Y\rangle \text{ of } e_U(\Pi \setminus b_U(\Pi), X),\ R'_Y \not\subset R \setminus b_U(R). \tag{10}$$

Let us prove that (8) implies (9). By contradiction, suppose that, for every candidate answer set $\langle S', R'\rangle$ of Π, $R' \not\subset R$, but that there exists a candidate answer set $\langle X', R'_X\rangle$ of $b_U(\Pi)$ such that $R'_X \subset b_U(R)$. Let (X'_D, X'_I) be the partition of X' such that X'_D is the set of the literals from X' that are relevant to the cr-rules in $R'_X \cup b_U(R)$. Let (X_D, X_U) be a similar partition of X and $X^{\sim} = X_I \cup X'_D$. From Definition 4, it is not difficult to prove that $\langle X^{\sim}, R'_X\rangle$ is a candidate answer set of $b_U(\Pi)$. Furthermore, $e_U(\Pi \setminus b_U(\Pi), X) = e_U(\Pi \setminus b_U(\Pi), X^{\sim})$. Hence, $\langle Y, R \setminus b_U(R)\rangle$ is a candidate answer set of $e_U(\Pi \setminus b_U(\Pi), X^{\sim})$.

Notice that $R \setminus b_U(R) = (R'_X \cup (R \setminus b_U(R))) \setminus b_U(R'_X)$, and that $b_U(R'_X) = b_U(R'_X \cup (R \setminus b_U(R)))$. Therefore, $\langle Y, R \setminus b_U(R)\rangle = \langle Y, (R'_X \cup (R \setminus b_U(R))) \setminus b_U(R'_X \cup (R \setminus b_U(R)))\rangle$, which allows us to conclude that $\langle Y, (R'_X \cup (R \setminus b_U(R))) \setminus b_U(R'_X \cup (R \setminus b_U(R)))\rangle$ is a candidate answer set of $e_U(\Pi \setminus b_U(\Pi), X^{\sim})$. By Lemma 2, $\langle X^{\sim} \cup Y, R'_X \cup (R \setminus b_U(R))\rangle$ is a candidate answer set of Π. Because $R'_X \subset b_U(R)$ by hypothesis, $R'_X \cup (R \setminus b_U(R)) \subset R$, which contradicts the assumption that, for every candidate answer set $\langle S', R'\rangle$ of Π, $R' \not\subset R$.

Let us now prove that (8) implies (10). By contradiction, suppose that, for every candidate answer set $\langle S', R'\rangle$ of Π, $R' \not\subset R$, but that there exists a candidate answer set $\langle Y', R'_Y\rangle$ of $e_U(\Pi \setminus b_U(\Pi), X)$ such that $R'_Y \subset R \setminus b_U(R)$. Because $R'_Y \subset R \setminus b_U(R)$, from Lemma 2 we conclude that $\langle X \cup Y', b_U(R) \cup R'_Y\rangle$ is a candidate answer set of Π, and that $b_U(R) \cup R'_Y \subset R$. But the hypothesis was that, for every candidate answer set $\langle S', R'\rangle$ of Π, $R' \not\subset R$. *Contradiction.*

Let us now prove that (9) and (10) imply (8). By contradiction, suppose that, for every candidate answer set $\langle S'_X, R'_X\rangle$ of $b_U(\Pi)$, $R'_X \not\subset b_U(R)$, and, for every candidate answer set $\langle S'_Y, R'_Y\rangle$ of $e_U(\Pi \setminus b_U(\Pi), X)$, $R'_Y \not\subset R \setminus b_U(R)$, but that there exists a candidate answer set $\langle S', R'\rangle$ of Π such that $R' \subset R$. By Lemma 2, (i) $\langle S' \cap U, b_U(R')\rangle$ is a candidate answer set of $b_U(\Pi)$, and $\langle S' \setminus U, R' \setminus b_U(R')\rangle$ is a candidate answer set of $e_U(\Pi \setminus b_U(\Pi), S' \cap U)$. Notice that, because $R' \subset R$, either $b_U(R') \subset b_U(R)$ or $R' \setminus b_U(R') \subset R \setminus b_U(R)$.

Case 1: $b_U(R') \subset b_U(R)$. It follows that $\langle S' \cap U, b_U(R') \rangle$ is a candidate answer set of $b_U(\Pi)$ such that $b_U(R') \subset b_U(R)$. *Contradiction.*

Case 2: $R' \setminus b_U(R') \subset R \setminus b_U(R)$. Let $X' = S' \cap U$, and (X'_I, X'_D) be the partition of X' such that X'_D consists of all of the literals of X that are relevant to the cr-rules in $b_U(R') \cup b_U(R)$. Let (X_I, X_D) be a similar partition of X. From Definition 4, it is not difficult to prove that $\langle S' \setminus U, R' \setminus b_U(R') \rangle$ is a candidate answer set of $e_U(\Pi \setminus b_U(\Pi), X_I \cup X'_D)$. Moreover, $e_U(\Pi \setminus b_U(\Pi), X_I \cup X'_D) = e_U(\Pi \setminus b_U(\Pi), X)$. Therefore, $\langle S' \setminus U, R' \setminus b_U(R') \rangle$ is a candidate answer set of $e_U(\Pi \setminus b_U(\Pi), X)$, and $R' \setminus b_U(R') \subset R \setminus b_U(R)$. This violates the assumption that, for every candidate answer set $\langle S'_Y, R'_Y \rangle$ of $e_U(\Pi \setminus b_U(\Pi), X)$, $R'_Y \not\subset R \setminus b_U(R)$. *Contradiction.*

References

1. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Proceedings of ICLP 1988, pp. 1070–1080 (1988)
2. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Computing, 365–385 (1991)
3. Marek, V.W., Truszczynski, M.: Stable models and an alternative logic programming paradigm. In: The Logic Programming Paradigm: a 25-Year Perspective, pp. 375–398. Springer, Berlin (1999)
4. Gelfond, M.: Representing Knowledge in A-Prolog. In: Kakas, A.C., Sadri, F. (eds.) Computational Logic: Logic Programming and Beyond, Part II. LNCS (LNAI), vol. 2408, pp. 413–451. Springer, Heidelberg (2002)
5. Baral, C.: Knowledge Representation, Reasoning, and Declarative Problem Solving. Cambridge University Press, Cambridge (2003)
6. Balduccini, M., Gelfond, M.: Logic Programs with Consistency-Restoring Rules. In: Doherty, P., McCarthy, J., Williams, M.A. (eds.) International Symposium on Logical Formalization of Commonsense Reasoning. AAAI 2003 Spring Symposium Series, March 2003, pp. 9–18 (2003)
7. Balduccini, M.: USA-Smart: Improving the Quality of Plans in Answer Set Planning. In: Jayaraman, B. (ed.) PADL 2004. LNCS (LNAI), vol. 3057, pp. 135–147. Springer, Heidelberg (2004)
8. Baral, C., Gelfond, M., Rushton, N.: Probabilistic reasoning with answer sets. Journal of Theory and Practice of Logic Programming (TPLP) (2005)
9. Balduccini, M., Gelfond, M., Nogueira, M.: Answer Set Based Design of Knowledge Systems. Annals of Mathematics and Artificial Intelligence (2006)
10. Son, T.C., Sakama, C.: Negotiation Using Logic Programming with Consistency Restoring Rules. In: 2009 International Joint Conferences on Artificial Intelligence, IJCAI (2009)
11. Lifschitz, V., Turner, H.: Splitting a logic program. In: Proceedings of the 11th International Conference on Logic Programming (ICLP 1994), pp. 23–38 (1994)
12. Balduccini, M., Mellarkod, V.S.: CR-Prolog with Ordered Disjunction. In: International Workshop on Non-Monotonic Reasoning, NMR 2004 (June 2004)
13. Balduccini, M.: CR-MODELS: An Inference Engine for CR-Prolog. In: Baral, C., Brewka, G., Schlipf, J. (eds.) LPNMR 2007. LNCS (LNAI), vol. 4483, pp. 18–30. Springer, Heidelberg (2007)
14. Turner, H.: Splitting a Default Theory. In: Proceedings of AAAI 1996, pp. 645–651 (1996)
15. Watson, R.: A Splitting Set Theorem for Epistemic Specifications. In: Proceedings of the 8th International Workshop on Non-Monotonic Reasoning (NMR 2000) (April 2000)