Simplified Reduct for Choice Rules in ASP

Mark Law, Alessandra Russo and Krysia Broda

April 25, 2015

Abstract

The accepted definitions of the semantics of choice rules in Answer Set Programming (ASP) involve a translation by inventing new atoms which do not occur in the original program. In this report, we put forward a new definition of the reduct for programs containing choice rules which does not invent new predicates and prove that the semantics are the same for the subset of ASP which we consider.

1 Background

1.1 Syntax

In this paper, we consider only a subset of the ASP language; specifically, we only consider aggregates when they occur in the head, and only when they are counting aggregates. We also do not consider conditionals or disjunction. The full class of programs we consider is formalised by the following definition.

Definition 1. We denote the class of ASP rules we consider as $\mathcal{L}$

For any ground atoms $\mathtt{h}_1, \ldots, \mathtt{h}_m, \mathtt{b}_1, \ldots, \mathtt{b}_n, \mathtt{c}_1, \ldots, \mathtt{c}_o$ and integers $\mathtt{l}$ and $\mathtt{u}$.

1. $\mathtt{h}_1 \texttt{ :- } \mathtt{b}_1, \ldots, \mathtt{b}_n, \texttt{not } \mathtt{c}_1, \ldots, \texttt{not } \mathtt{c}_o \in \mathcal{L}$
2. $\texttt{:- } \mathtt{b}_1, \ldots, \mathtt{b}_n, \texttt{not } \mathtt{c}_1, \ldots, \texttt{not } \mathtt{c}_o \in \mathcal{L}$
3. $\mathtt{l}\{\mathtt{h}_1; \ldots; \mathtt{h}_m\}\mathtt{u} \texttt{ :- } \mathtt{b}_1, \ldots, \mathtt{b}_n, \texttt{not } \mathtt{c}_1, \ldots, \texttt{not } \mathtt{c}_o \in \mathcal{L}$

No other rule is in $\mathcal{L}$.

Note that as is conventional, we define our semantics over ground programs; we assume that non-ground programs are "grounded" as described in [1]. We consider any ASP program whose relevant grounding is a finite subset of $\mathcal{L}$.

We must also define rules with disjunction in the head, as they are used in the conventional semantics for choice rules.

Definition 2. For any ground atoms $\mathtt{h}_1, \ldots, \mathtt{h}_m, \mathtt{b}_1, \ldots, \mathtt{b}_n, \mathtt{c}_1, \ldots, \mathtt{c}_o$ and integers $\mathtt{l}$ and $\mathtt{u}$.

$\mathtt{h}_1 \vee \ldots \vee \mathtt{h}_m \texttt{ :- } \mathtt{b}_1, \ldots, \mathtt{b}_n, \texttt{not } \mathtt{c}_1, \ldots, \texttt{not } \mathtt{c}_o \notin \mathcal{L}$, but is a rule which we consider in the report.

1.2 Semantics

In this section we give the standard, accepted definitions of the semantics of ASP programs. In this report, we do not consider all ASP constructs; we only consider programs with normal rules, choice rules and constraints.

Definition 3. Given a program P:

1. The Herbrand base (written HB_P) is the set of all atoms which can be constructed from the predicate names, function symbols and constant symbols in P.

2. An Herbrand interpretation assigns each element of HB_P to be either true or false. We denote an interpretation I as the subset of HB_P which I assigns to true.
3. An atom $\mathtt{a}$ is satisfied by an interpretation I if $\mathtt{a} \in I$
4. An aggregate $\mathtt{l\{h_1;\ldots;h_m\}u}$ is satisfied by an interpretation I if $l \leq |\{h_1, \ldots h_m\} \cap I| \leq u$.
5. For any rule R of the form $\mathtt{h_1} \vee \ldots \vee \mathtt{h_m}$:- $\mathtt{b_1}, \ldots, \mathtt{b_n}, \mathtt{not\ c_1}, \ldots, \mathtt{not\ c_o}$:

   body(R) is satisfied by an interpretation I if $\forall i \in \{1, \ldots, n\}$, $\mathtt{b_i}$ is satisfied by I and $\forall i \in \{1, \ldots, o\}$, $\mathtt{c_i}$ is not satisfied by I.

   head(R) is satisfied by an interpretation I is $\exists i \in \{1, \ldots m\}$ st $\mathtt{h_i} \in I$.
6. If P has no aggregates in the heads of rules then an Herbrand interpretation is an Herbrand model of P if for every rule $R \in P$ such that *body*(R) is satisfied, *head*(R) is also satisfied. We will write *models*(P) to denote the set of all Herbrand models of P.
7. An Herbrand interpretation I is a minimal Herbrand model if it is an Herbrand model and no strict subset of I is also an Herbrand model.

   We write $M(P)$ to denote the set of minimal Herbrand models of P.
8. For a program P with no aggregates in the heads of rules, the reduct of P with respect to an Herbrand interpretation I is the program consisting of all rules in P whose bodies are satisfied.
9. For a program P with no aggregates in the heads of rules, an Herbrand interpretation I is an answer set of a program P if it is a minimal herbrand model of the reduct of P with respect to I.

In [1], the semantics of choice rules is defined by a translation to normal rules. The application of their translation to our subset of ASP is formalised by the following definition.

Definition 4. We define *translate* to be a mapping from any subset of $\mathcal{L}$ to an ASP program.

- Given a choice rule R:

  $\mathtt{l\{h_1;\ldots;h_m\}u}$:- $\mathtt{b_1}, \ldots, \mathtt{b_n}, \mathtt{not\ c_1}, \ldots, \mathtt{not\ c_o}$.

  $$translate(R) = \left\{ \begin{array}{c} \mathtt{h_1} \vee \widehat{\mathtt{h_1}} \mathtt{:-\ b_1}, \ldots, \mathtt{b_n}, \mathtt{not\ c_1}, \ldots, \mathtt{not\ c_o}. \\ \ldots \\ \mathtt{h_m} \vee \widehat{\mathtt{h_m}} \mathtt{:-\ b_1}, \ldots, \mathtt{b_n}, \mathtt{not\ c_1}, \ldots, \mathtt{not\ c_o}. \\ \mathtt{:-\ b_1}, \ldots, \mathtt{b_n}, \mathtt{not\ c_1}, \ldots, \mathtt{c_o}, \mathtt{not\{h_1}, \ldots, \mathtt{h_m\}u}. \\ \mathtt{:-\ b_1}, \ldots, \mathtt{b_n}, \mathtt{not\ c_1}, \ldots, \mathtt{c_o}, \mathtt{not\ l\{h_1}, \ldots, \mathtt{h_m\}}. \end{array} \right\}$$

  where $\widehat{h_i}$ are new atoms which occur nowhere else in the program. They essentially represent the complement of the atom.
- Given any other (non-choice) rule R, $translate(R) = R$.

To cut down on excessive notation we define the reduct of the translation of a program P with respect to an interpretation I. To avoid confusion with our own later definition of the reduct, we denote this P^I_{old}.

Definition 5. Given an ASP program P and an interpretation I, the old reduct of P with respect to I, written P^I_{old}, is constructed in two steps:

1. Replace all rules $R \in P$ with $translate(R)$
2. Remove any rule whose body is not satisfied by R

Although this is the accepted definition of a reduct for a program including choice rules, it has, in our opinion, some undesirable properties: firstly, it involves inventing new atoms which are not in HB_P; secondly, it does not guarantee that the reduct has a unique minimal model. This is because the reduct is not, as was originally the case in [2], a definite logic program. This means that it can be unintuitive to those who are comfortable with the original definition of a reduct.

An answer set of P does not include the extra $\widehat{h_i}$ predicates. The following definition extends the concept of answer set to programs including choice rules. We write AS_{old} to denote the set of answer sets, under this semantics, of a program P.

Definition 6. Given a program P and an interpretation I of P, I' is an *extension* of I if and only if $I = (I' \cap HB_P)$.

Given a program P and an interpretation I, $I \in AS_{old}(P)$ if and only if $\exists I' \in M(P^I_{old})$ such that I' is an extension of I.

Example 1.1. Consider the program $P = \left\{ \begin{array}{l} \texttt{1\{p;q\}1 :- r.} \\ \texttt{r.} \\ \texttt{:- not p, r.} \end{array} \right\}$

The reduct wrt the interpretation $I = \{p, r\}$ is $\left\{ \begin{array}{l} \texttt{p} \vee \widehat{\texttt{p}} \texttt{ :- r.} \\ \texttt{q} \vee \widehat{\texttt{q}} \texttt{ :- r.} \\ \texttt{:- r, not\{p;q\}1.} \\ \texttt{:- r, not 1\{p;q\}.} \\ \texttt{r.} \end{array} \right\}$

The minimal models of the reduct are $\{p, \widehat{q}, r\}$ and $\{\widehat{p}, q, r\}$. Hence, as $\{p, \widehat{q}, r\} \cap HB_P = I$, I is an answer set of P.

Consider now the alternative interpretation $I' = \{q, r\}$.

The reduct wrt I' is $\left\{ \begin{array}{l} \texttt{p} \vee \widehat{\texttt{p}} \texttt{ :- r.} \\ \texttt{q} \vee \widehat{\texttt{q}} \texttt{ :- r.} \\ \texttt{:- r, not\{p;q\}1.} \\ \texttt{:- r, not 1\{p;q\}.} \\ \texttt{r.} \\ \texttt{:- not p, r.} \end{array} \right\}$

There is now only one minimal model of the reduct (as $\{\widehat{p}, q, r\}$ violates the constraint). This model is: $\{p, \widehat{q}, r\}$. Hence, as $\{p, \widehat{q}, r\} \cap HB_P \neq I'$, I' is not an answer set of P.

Consider a third interpretation $I'' = \{p, q, r\}$.

The reduct wrt I' is $\left\{ \begin{array}{l} \texttt{p} \vee \widehat{\texttt{p}} \texttt{ :- r.} \\ \texttt{q} \vee \widehat{\texttt{q}} \texttt{ :- r.} \\ \texttt{:- r, not\{p;q\}1.} \\ \texttt{:- r, not 1\{p;q\}.} \\ \texttt{r.} \end{array} \right\}$

The minimal models of the reduct are $\{p, \widehat{q}, r\}$ and $\{\widehat{p}, q, r\}$. Hence, as neither contains both p and q, I'' is not an answer set of P.

2 Simplified Semantics

In this section, we present our simplified semantics for programs in $\mathcal{L}$. This does not involve a translation before solving; instead it uses an extended definition of reduct. The motivation for this is *not* to change the semantics, improve solving or even to change the standard intuition of these rules; it is to provide alternative definitions which can simplify proofs of properties about an ASP program. This is achieved by removing the need to "invent" extra helper predicates which do not appear in the answer sets of a program.

Definition 7. The reduct of a program P with respect to an interpretation I, is constructed in the following 4 steps.

1. Remove any rule whose body contains `not a` for some $\texttt{a} \in I$ and remove any negative literals from the remaining rules.
2. For any constraint R, `:-`$body(R)$, replace R with $\bot$ `:-`$body^+(R)$ ($\bot$ is a new atom which cannot appear in any answer set of P).
3. For any choice rule R, $\texttt{l\{h}_1\texttt{;}\ldots\texttt{;h}_n\texttt{\}u :-}body(R)$ such that $l \leq |I \cap \{h_1, \ldots, h_n\}| \leq u$, replace R with the set of rules $\{\texttt{h}_\texttt{i}\texttt{ :-}body^+(R) \mid h_i \in I \cap \{h_1 \ldots h_n\}\}$.
4. For any remaining choice rule R, $\texttt{l\{h}_1\texttt{;}\ldots\texttt{;h}_n\texttt{\}u :-}body(R)$, replace R with the constraint $\bot$ `:-`$body^+(R)$.

Example 2.1. Consider the program P from example 1.1, $\left\{ \begin{array}{l} \texttt{1\{p;q\}1 :- r.} \\ \texttt{r.} \\ \texttt{:- not p, r.} \end{array} \right\}$

The reduct wrt the interpretation $I = \{p, r\}$ is $\left\{ \begin{array}{l} \texttt{p :- r.} \\ \texttt{r.} \end{array} \right\}$

The minimal modelsof the reduct is I, and hence, I is an answer set of P.

Consider now the alternative interpretation $I' = \{q, r\}$.

The reduct wrt I' is $\left\{ \begin{array}{l} \texttt{q :- r.} \\ \texttt{r.} \\ \bot \texttt{ :- r.} \end{array} \right\}$

The minimal model of the reduct is: $\{q, r, \bot\}$. Hence, as $\bot \notin I'$, $I' \notin AS(P)$.

Consider the third interpretation $I'' = \{p, q, r\}$.

The reduct wrt I' is $\left\{ \begin{array}{l} \bot \texttt{ :- r.} \\ \texttt{r.} \end{array} \right\}$

The minimal model is $\{r, \bot\}$, and hence, as $\bot \notin I''$, $I'' \notin AS(P)$.

3 Proof of Equivalence

In this section we show that for any program $P \subset \mathcal{L}$, $AS(P) = AS_{old}(P)$; hence, our new simplified definitions of the semantics of ASP are equivalent over the language of $\mathcal{L}$.

Before proving this result, we prove some results about the reducts of single rules in a program. The intuition is that as for any interpretation I and program P, $P^I = \{R^I \mid R \in P\}$ and hence $models(P^I) = \bigcap_{R \in P} models(R^I)$ (similarly for P^I_{old}), we can use these results to reason about the models of the whole reduct.

Lemmas 3.1 and 3.2 state that when R is a normal rule (respectively a constraint) and I is any interpretation, any subset of I is a model of R^I if and only if it is a model of R^I_{old}. This property is useful when reasoning about whether I is a model of the reduct and then whether it is a minimal model.

Lemma 3.1. Given any program P and interpretation I, for any $I' \subseteq I$:

Let R be any normal rule; then $I' \in models(R^I_{old})$ if and only if $I' \in models(R^I)$

Proof. We prove this result by showing instead the equivalent result: $I' \notin models(R^I_{old}) \Leftrightarrow I' \notin models(R^I)$.

(Proof of $\Leftarrow$):

Let R be any normal rule, $I' \subseteq I$ and assume that $I' \notin models(R^I)$

Assume for contradiction that I does not satisfy the body of R.

Case 1: I does not satisfy $body^-(R)$

$\Rightarrow R^I = \emptyset$

$\Rightarrow I' \in models(R^I)$. Contradiction!

Case 2: I satisfies $body^-(R)$ and I does not satisfy $body^+(R)$

$\Rightarrow I'$ does not satisfy $body^+(R)$ (as $I' \subset I$)

But $R^I = \{head(R) \leftarrow body^+(R)\}$

So $I' \in models(R^I)$. Contradiction!

Hence as both cases resulted in contradictions, I must satisfy the body of R.

$\Rightarrow R^I = \{head(R) \leftarrow body^+(R)\}$ and $R^I_{old} = R$ (as I satisfies the body of R) and I' satisfies $body^-(R)$ (as $I' \subseteq I$).

$\Rightarrow I'$ satisfies $body^+(R)$ but not $head(R)$ (as I' is not a model of R^I).

$\Rightarrow I'$ does not satisfy R (as it satisfies the body but not the head).

$\Rightarrow I' \notin models(R^I_{old})$.

(Proof of $\Rightarrow$):

Let R be any normal rule, $I' \subseteq I$ and assume that $I' \notin R^I_{old}$

Assume for contradiction that I does not satisfy the body of R.

$\Rightarrow R^I_{old} = \emptyset$

$\Rightarrow I' \in models(R^I_{old})$. Contradiction!

Hence I satisfies the body of R.

$\Rightarrow R^I_{old} = \{R\}$ and $R^I = \{head(R) \leftarrow body^+(R)\}$

$\Rightarrow I'$ satisfies $body(R)$ but not $head(R)$.

$\Rightarrow I' \notin models(R^I)$ (as I' does not satisfy the single rule in R^I)

□

Lemma 3.2 states the same result as Lemma 3.1 but for constraints. The proof is similar.

Lemma 3.2. Given any program P and interpretation I, for any $I' \subseteq I$:

Let R be any constraint; then $I' \in models(R^I_{old})$ if and only if $I' \in models(R^I)$

Proof. Again, we instead prove the equivalent result: $I' \notin models(R^I_{old}) \Leftrightarrow I' \notin models(R^I)$.

(Proof of $\Leftarrow$):

Let R be any constraint, $I' \subseteq I$ and assume that $I' \notin models(R^I)$

Assume for contradiction that I does not satisfy the body of R.

Case 1: I does not satisfy $body^-(R)$

$\Rightarrow R^I = \emptyset$

$\Rightarrow I' \in models(R^I)$. Contradiction!

Case 2: I satisfies $body^-(R)$ and I does not satisfy $body^+(R)$

$\Rightarrow I'$ does not satisfy $body^+(R)$ (as $I' \subset I$)

But $R^I = \{\bot \leftarrow body^+(R)\}$

So I' must satisfy R^I. Contradiction!

Hence, as both cases resulted in contradictions, I must satisfy the body of R.

$\Rightarrow R^I_{old} = R$ and $R^I = \{\bot$:- $body^+(R)\}$ (as I satisfies the body of R)

$\Rightarrow I'$ satisfies $body^+(R)$ (as $I' \notin models(R^I)$

$\Rightarrow I' \notin models(R^I_{old})$ (as I' satisfies the body of a constraint in R^I_{old}).

(Proof of $\Rightarrow$):

Let R be any constraint rule, $I' \subseteq I$ and assume that $I' \notin R^I_{old}$

Assume for contradiction that I does not satisfy the body of R.

$\Rightarrow R^I_{old} = \emptyset$

$\Rightarrow I' \in models(R^I_{old})$. Contradiction!

Hence I satisfies the body of R.

$\Rightarrow R^I_{old} = R$ and $R^I = \{\bot \texttt{ :- } body^+(R)\}$ (as I satisfies the body of R)

$\Rightarrow I'$ satisfies $body^+(R)$ (as $I' \notin models(R^I_{old})$

$\Rightarrow I' \notin models(R^I)$ (as $\bot \notin I'$)

□

Lemma 3.3. Given any program P and interpretation I:

Let R be any choice rule in P; then $I \in models(R^I)$ if and only if there is an extension of I, I_{ext} st $I_{ext} \in models(R^I_{old})$

Proof.

Let R be the choice rule $\texttt{lower}\{\texttt{h}_1;\ldots;\texttt{h}_\texttt{m}\}\texttt{upper} \texttt{ :- } \texttt{b}_1,\ldots,\texttt{b}_\texttt{n},\texttt{not c}_1,\ldots,\texttt{not c}_\texttt{o}$.

Case 1: I satisfies $b_1,\ldots,b_n,\texttt{not}\, c_1,\ldots,\texttt{not}\, c_o$

We prove the equivalent result that $I \notin models(R^I) \Leftrightarrow$ there is no extension I_{ext} of I st $I_{ext} \in R^I_{old}$.

$$R^I_{old} = \left\{ \begin{array}{c} \texttt{h}_1 \vee \widehat{\texttt{h}_1} \texttt{ :- } \texttt{b}_1,\ldots,\texttt{b}_\texttt{n},\texttt{not c}_1,\ldots,\texttt{not c}_\texttt{o}. \\ \ldots \\ \texttt{h}_\texttt{m} \vee \widehat{\texttt{h}_\texttt{m}} \texttt{ :- } \texttt{b}_1,\ldots,\texttt{b}_\texttt{n},\texttt{not c}_1,\ldots,\texttt{not c}_\texttt{o}. \\ \texttt{:- b}_1,\ldots,\texttt{b}_\texttt{n},\texttt{not c}_1,\ldots,\texttt{not c}_\texttt{o},\texttt{not}\{\texttt{h}_1;\ldots;\texttt{h}_\texttt{m}\}\texttt{upper}. \\ \texttt{:- b}_1,\ldots,\texttt{b}_\texttt{n},\texttt{not c}_1,\ldots,\texttt{not c}_\texttt{o},\texttt{not lower}\{\texttt{h}_1;\ldots;\texttt{h}_\texttt{m}\}. \end{array} \right\}$$

Assume $I \notin models(R^I)$

Assume for contradiction that I satisfies $\texttt{lower}\{\texttt{h}_1;\ldots;\texttt{h}_\texttt{m}\}\texttt{upper}$

$\Rightarrow R^I = \{\texttt{h}_\texttt{i} \leftarrow \texttt{b}_1,\ldots,\texttt{b}_\texttt{n}. \mid h_i \in \{h_1,\ldots,h_m\} \cap I\}$

$\Rightarrow$ the head of every rule in R^I is satisfied by I.

$\Rightarrow I \in models(R^I)$. Contradiction!

So $I \notin models(R^I) \Rightarrow I$ does not satisfy $\texttt{lower}\{\texttt{h}_1;\ldots;\texttt{h}_\texttt{m}\}\texttt{upper}$.

But I does not satisfy $\texttt{lower}\{\texttt{h}_1;\ldots;\texttt{h}_\texttt{m}\}\texttt{upper} \Rightarrow R^I = \{\bot \texttt{ :- } \texttt{b}_1,\ldots,\texttt{b}_\texttt{n}\}$ which is not satisfied by I.

So $I \notin models(R^I) \Leftrightarrow I$ does not satisfy $\texttt{lower}\{\texttt{h}_1;\ldots;\texttt{h}_\texttt{m}\}\texttt{upper}$.

$\Leftrightarrow I \cap \{\texttt{h}_1,\ldots,\texttt{h}_\texttt{m}\} > upper$ or $I \cap \{\texttt{h}_1,\ldots,\texttt{h}_\texttt{m}\} < lower$

$\Leftrightarrow I$ satisfies $\texttt{not}\{\texttt{h}_1;\ldots;\texttt{h}_\texttt{m}\}\texttt{upper}$ or $\texttt{not lower}\{\texttt{h}_1;\ldots;\texttt{h}_\texttt{m}\}$

$\Leftrightarrow \forall I_{ext}$ st $I = (I_{ext} \cap HB_P)$, $I_{ext} \notin models(R^I_{old})$

(The $\Leftarrow$ holds because of the $\forall$; if the constraints were satisfied by an I_{ext} then $I_{ext} \cup \{\widehat{h_1},\ldots,\widehat{h_m}\}$ would be a model of R^I_{old}).

$\Leftrightarrow$ no extension of I is in $models(R^I_{old})$

$\therefore I \notin models(R^I) \Leftrightarrow$ no extension of I is in $models(R^I_{old})$.

Hence in this case, $I \in models(R^I) \Leftrightarrow$ there is an extension of I, I_{ext} st $I \in models(R^I_{old})$.

Case 2: I does not satisfy $\texttt{b}_1,\ldots,\texttt{b}_\texttt{n},\texttt{not c}_1,\ldots,\texttt{not c}_\texttt{o}$

Case 2a: I does not satisfy $\texttt{b}_1,\ldots,\texttt{b}_\texttt{n}$ but does satisfy $\texttt{not c}_1,\ldots,\texttt{not c}_\texttt{o}$

$I \in models(R^I)$ as the body of each rule in R^I is false. Similarly (as $R^I_{old} = \emptyset$), $I \in models(R^I_{old})$.

So as both $I \in models(R^I)$ and $\exists I_{ext}$ st $I = (I_{ext} \cap HB_P)$ and $I \in models(R^I_{old})$ share the same truth value in this case, $I \in models(R^I) \Leftrightarrow$ there is an extension of I, I_{ext} st $I_{ext} \in models(R^I_{old})$.

Case 2b: I does not satisfy $\texttt{not c}_1,\ldots,\texttt{not c}_\texttt{o}$

$R^I = R^I_{old} = \emptyset$.

So both $I \in models(R^I)$ and $I \in models(R^I_{old})$ are true.

Hence, in this case, $I \in models(R^I) \Leftrightarrow$ there is an extension of I, I_{ext} st $I_{ext} \in models(R^I_{old})$

Hence, as the cases are exhaustive, $I \in models(R^I) \Leftrightarrow$ there is an extension of I, I_{ext} st $I_{ext} \in models(R^I_{old})$

□

Lemma 3.4. Given any program P, interpretation I of P and choice rule R, $\mathtt{l\{h_1;\ldots;h_m\}\ :-\ b_1,\ldots,b_n,not\ c_1,\ldots,not\ c_o}$, such that $I \in models(R^I)$.

Let I_{ext} be an extension of I such that $I_{ext} \cap \{\widehat{atom} \mid atom \in I\} = \emptyset$ and $I_{ext} \in models(R^I_{old})$.

$\forall I' \subset I$, $I' \in models(R^I)$ if and only if there is an extension I'_{ext} of I' such that $I'_{ext} \in models(R^I_{old})$ and $I'_{ext} \subset I_{ext}$.

Proof.

First we show ($\Leftarrow$):

Let $I' \subset I$ and assume there is an extension of I', I'_{ext}, in $models(R^I_{old})$

Case 1: $body(R)$ is satisfied by I

$$R^I_{old} = \left\{ \begin{array}{c} \mathtt{h_1} \vee \widehat{\mathtt{h_1}}\ \mathtt{:-\ b_1,\ldots,b_n,not\ c_1,\ldots,not\ c_o.} \\ \ldots \\ \mathtt{h_m} \vee \widehat{\mathtt{h_m}}\ \mathtt{:-\ b_1,\ldots,b_n,not\ c_1,\ldots,not\ c_o.} \\ \mathtt{:-\ b_1,\ldots,b_n,not\ c_1,\ldots,not\ c_o,not\{h_1;\ldots;h_m\}upper.} \\ \mathtt{:-\ b_1,\ldots,b_n,not\ c_1,\ldots,not\ c_o,not\ lower\{h_1;\ldots;h_m\}.} \end{array} \right\}$$

Case 1a: $\{h_1,\ldots,h_m\} \cap I' \subset \{h_1,\ldots,h_m\} \cap I$

Assume for contradiction that $body(R)$ is satisfied by I'_{ext}.

$\Rightarrow \forall h_i \in \{h_1,\ldots,h_m\}, h_i \in I'_{ext} \vee \widehat{h_i} \in I'_{ext}$ as I'_{ext} is a model of R^I_{old}.

$\Rightarrow \exists \widehat{h_i} \in I'_{ext}$ st $\widehat{h_i} \notin I_{ext}$ (as $\exists h_i \in I$ st $h_i \notin I'$ because $I' \cap \{h_1,\ldots,h_m\} \subset I \cap \{h_1,\ldots,h_m\}$ and h_i and $\widehat{h_i}$ cannot both be in I_{ext}).

Contradiction as $I'_{ext} \subset I_{ext}$.

Hence $body(R)$ is not satisfied by I'_{ext}

$\Rightarrow I'$ does not satisfy $body(R)$ (as $(I'_{ext} \cap HB_P) = I'$).

$\Rightarrow I' \in models(R^I)$

Case 1b: $\{h_1,\ldots,h_m\} \cap I' = \{h_1,\ldots,h_m\} \cap I$

Case 1bi: $body(R)$ is satisfied by I'

$\Rightarrow body(R)$ is satisfied by I'_{ext}

$\Rightarrow$ (by the two constraints in R^I_{old}) $head(R)$ is satisfied by I'_{ext} and hence I_{ext} (as they share the same h_i's).

$\Rightarrow R^I = \{\mathtt{h_i\ :-}body(R) \mid h_i \in \{h_1,\ldots,h_m\} \cap I\}$

$\Rightarrow I' \in models(R^I)$ as I' shares all the h_i's in I and therefore satisfies the head of every rule in R^I.

Case 1bii: I' does not satisfy $body(R)$

$\Rightarrow I'$ does not satisfy $body^+(R)$ (as $I' \subset I$ and I satisfies $body^-(R)$, I' must satisfy $body^-(R)$).

$\Rightarrow I'$ satisfies the body of every rule in R^I

$\Rightarrow I' \in models(R^I)$

Case 2: $body^-(R)$ is not satisfied by I

$\Rightarrow R^I = \emptyset$ and hence $I' \in models(R^I)$.

Case 3: $body^-(R)$ is satisfied by I but $body^+(R)$ is not.

$\Rightarrow body^+(R)$ not satisfied by I' as $(I' \subseteq I)$

$\Rightarrow I' \in models(R^I)$ as $body^+(R)$ is the body of every rule in R^I.

Next we show ($\Rightarrow$):

Let I' be any interpretation such that $I' \subset I$

Assume $I' \in models(R^I)$

Case 1: $body(R)$ is satisfied by I

$\Rightarrow body^-(R)$ is satisfied by I' (as $I' \subset I$).

Case 1a: $body^+(R)$ is satisfied by I'

Assume for contradiction that $head(R)$ is not satisfied by I.

$\Rightarrow R^I = \{\bot \texttt{ :-} body^+(R).\}$

$\Rightarrow I' \notin models(R^I)$

Contradiction!

$\therefore head(R)$ is satisfied by I

$\Rightarrow R^I = \{\texttt{h}_\texttt{i} \texttt{ :-} body(R) \mid h_i \in \{h_1, \ldots h_m\} \cap I\}$

$\Rightarrow I \cap \{h_1, \ldots, h_m\} = I' \cap \{h_1, \ldots, h_m\}$

$\Rightarrow I'_{ext} = (I' \cup (I_{ext} \backslash HB_P))$ satisfies R^I_{old} (as $I_{ext} \cap HB_{R^I_{old}} = I'_{ext} \cap HB_{R^I_{old}}$).

$\Rightarrow$ there is an extension I'_{ext} of I' such that $I'_{ext} \in models(R^I_{old})$

Case 1b: $body^+(R)$ is not satisfied by I'

$\Rightarrow I' \in models(R^I_{old}$ (as the body of each rule in R^I_{old} is false).

$\Rightarrow$ there is an extension I'_{ext} of I' such that $I'_{ext} \in models(R^I_{old})$

Case 2: $body(R)$ is not satisfied by I

$\Rightarrow R^I_{old} = \emptyset$

$\Rightarrow I' \in models(R^I_{old})$

$\Rightarrow$ there is an extension I'_{ext} of I' such that $I'_{ext} \in models(R^I_{old})$

Hence, as the cases were exhaustive, $\exists I'_{ext}$ such that $I' = (I'_{ext} \cap HB_P)$ and $I'_{ext} \in models(R^I_{old})$

□

The next two lemmas use the previous results on the different types of rules (normal rules, choice rules and constraints) to give general results about any rule in $\mathcal{L}$. Lemma 3.5 states that an interpretation I is a model of the new reduct of a rule with respect to I if and only if there is an extension of I which is a model of R^I_{old}.

Lemma 3.5. Given any program P, interpretation I and a rule $R \in P$

$I \in models(R^I)$ if and only if there is an extension of I which is in $models(R^I_{old})$.

Proof.

Case 1: R is a normal rule.

Follows from lemma 3.1.

Case 2: R is a constraint.

Follows from lemma 3.2.

Case 3: R is a choice rule.

Follows from lemma 3.3.

□

Lemma 3.6 will be useful for proving whether interpretations are minimal models of their reduct. It states that for any rule R, given an interpretation I which is a model of R^I and an extension I_{ext} of I such that I doesn't contain any atom a and its complement $\widehat{a}$, any subset I' of I is also a model of R^I if and only if there is a subset of I_{ext} which is an extension of I' which is a model of R^I_{old}. Lemma 3.4 states this explicitly when R is a choice rule, and the other two cases follow from lemmas 3.1 and 3.2.

Lemma 3.6. Given any program P, a rule $R \in P$ an interpretation $I \in models(R^I)$ and an extension of I, I_{ext}, such that $I_{ext} \in models(R^I_{old})$ and $I_{ext} \cap \{\widehat{atom} \mid atom \in I\} = \emptyset$,

$\forall I' \subset I$, $I' \in models(R^I)$ if and only if there is an extension of I', I'_{ext} st $I'_{ext} \subset I'_{ext}$ and $I'_{ext} \in models(R^I_{old})$

Proof. **Case 1:** R is a normal rule.

As R^I_{old} only contains atoms from the Herbrand Base of P:

$\exists$ an extension of I', I'_{ext} such that and $I'_{ext} \in models(R^I_{old})$ and $I'_{ext} \subset I_{ext} \Leftrightarrow I' \in models(R^I_{old})$

$\Leftrightarrow I' \in models(R^I)$ (by lemma 3.1).

Case 2: R is a constraint.

As R^I_{old} only contains atoms from the Herbrand Base of P:

$\exists$ an extension of I', I'_{ext} such that and $I'_{ext} \in models(R^I_{old})$ and $I'_{ext} \subset I_{ext} \Leftrightarrow I' \in models(R^I_{old})$

$\Leftrightarrow I' \in models(R^I)$ (by lemma 3.2).

Case 3: R is a choice rule.

Follows from lemma 3.4.

□

Theorem 1. Given any ASP program P, an interpretation A is in $AS_{old}(P)$ if and only if $A \in AS(P)$.

Proof.

1. Proof of $\Rightarrow$:

   Assume $I \in AS_{old}(P)$

   $\Rightarrow$ There is an extension I_{ext} of I which is in $M(P^I_{old})$

   Note that there cannot be any atom $a \in I$ such that $\widehat{a} \in I_{ext}$ (as without $\widehat{a}$ this would still be a model, breaking minimality).

   $\Rightarrow$ There is an extension I_{ext} of I st $I_{ext} \in models(P^I_{old})$ and $\forall I'_{ext} \subset I_{ext}$, $I'_{ext} \notin models(P^I_{old})$ and $I_{ext} \cap \{\widehat{a} \mid a \in I\} = \emptyset$.

   $\Rightarrow$ There is an extension I_{ext} of I st $(\forall R \in P, I_{ext} \in models(R^I_{old}))$ and $\forall I'_{ext} \subset I_{ext}$, $I'_{ext} \notin models(P^I_{old})$ and $I_{ext} \cap \{\widehat{a} \mid a \in I\} = \emptyset$.

   $\Rightarrow (\forall R \in P, I \in models(R^I))$ and there is an extension I_{ext} of I st $\forall I'_{ext} \subset I_{ext}$, $I'_{ext} \notin models(P^I_{old})$ and $I_{ext} \cap \{\widehat{a} \mid a \in I\} = \emptyset$ by lemma 3.5.

   Assume for contradiction that $\exists I' \subset I$ such that $\forall R \in P$, $I' \in models(R^I)$

   $\Rightarrow \forall R \in P$ there is an extension I^R_{ext} of I' such that $I^R_{ext} \subset I_{ext}$ and $I^R_{ext} \in models(R^I_{old})$ (by lemma 3.6)

   $\Rightarrow$ there is an extension I^*_{ext} of I' such that $\forall R \in P, I^*_{ext} \in models(R^I_{old})$ and $I^*_{ext} \subset I_{ext}$ (the union of the previous I^R_{ext}'s is a model as it makes no bodies true which weren't true in all the I^R_{ext}'s as the bodies only contain atoms from the Herbrand Base).

   Contradiction as $\forall I'_{ext} \subset I_{ext}$, $I'_{ext} \notin models(P^I)$.

   $\therefore (\forall R \in P, I \in models(R^I))$ and I_{ext} of I st $\forall I' \subset I$, $\exists R \in P$ st $I' \notin models(R^I)$.

   $\Rightarrow (\forall R \in P, I \in models(R^I))$ and $\forall I' \subset I$, $I' \notin models(P^I)$

   $\Rightarrow I \in models(P^I)$ and $\forall I' \subset I$, $I' \notin models(P^I)$

   $\Rightarrow I \in M(P^I)$

   $\Rightarrow I \in AS(P)$

2. Proof of $\Leftarrow$:

   Assume $I \in AS(P)$

   $\Rightarrow I \in M(P^I)$

   $\Rightarrow I \in models(P^I)$

   $\Rightarrow \forall R \in P : I \in models(R^I)$

   $\Rightarrow \forall R \in P$, there is an extension of I, I^R_{ext} st $I^R_{ext} \in models(R^I_{old})$ (by lemma 3.5).

   As for each I^R_{ext}, $I^R_{ext} \cap HB_P = I$, the only elements which differ in these extended interpretations do not occur in the bodies of any rule in R^I_{old}. Hence, the union of all I_{ext}'s does not satisfy the body of any rule which was not satisfied by I.

   Hence, for each rule R in the reduct P^I_{old} whose body is satisfied by I, there is at least one I^R_{ext} which satisfies $head(R)$. Hence the union of all I^R_{ext}'s must satisfy $head(R)$.

   Hence this union is a model of P^I_{old} which is an extension of I.

   So there is an extension of I which is a model of P^I_{old}. Let I^*_{ext} be the smallest such extension (this means that $I^*_{ext} \cap \{\widehat{a} \mid I\} = \emptyset$).

   Assume for contradiction that $\exists I'_{ext}$ st $I'_{ext} \subset I^*_{ext}$ and $I'_{ext} \in models(P^I_{old})$.

      Let $I' = (I'_{ext} \cap HB_P)$ (So I'_{ext} is an extension of I')

      $\Rightarrow I' \subset I$ (as I^*_{ext} was the smallest extension of I in $models(P^I_{old}$ and $I'_{ext} \in models(P^I_{old}))$

      As $I'_{ext} \in models(P^I_{old})$, $\forall R \in P$, $I'_{ext} \in models(R^I_{old})$.

      $\Rightarrow \forall R \in P$, $I' \in models(R^I)$ (by lemma 3.6).

      $\Rightarrow I' \in models(P^I)$

      Contradiction! As I was a minimal model of P^I.

   $\Rightarrow$ there is an extension I_{ext} of I st $I_{ext} \in models(P^I_{old})$ and $\forall I'_{ext} \subset I_{ext}$, $I'_{ext} \notin models(P^I_{old})$

   $\Rightarrow$ there is an extension I_{ext} of I in $M(P^I_{old})$

   $\Rightarrow I \in AS_{old}(P)$

□

4 Conclusion and Future Work

In this report we presented a new simplified definition for the reduct of ASP programs consisting of normal rules, choice rules and constraints. We have shown that the semantics of these programs with this new definition of the reduct is equivalent to the accepted semantics of ASP.

We have not considered other types of ASP rules; for example, aggregates in the body, other types of aggregates and atoms with conditions. Future work could address extending our new definitions to incorporate the entire language of ASP.

References

[1] F. Calimeri, W. Faber, M. Gebser, G. Ianni, R. Kaminski, T. Krennwallner, N. Leone, F. Ricca, T. Schaub, Asp-core-2 input language format (2013).

[2] M. Gelfond, V. Lifschitz, The stable model semantics for logic programming., in: ICLP/SLP, Vol. 88, 1988, pp. 1070–1080.