A Semantic characterization for ASP base revision

Laurent Garcia, Claire Lefevre, Odile Papini, Igor Stephan, Eric Würbel

A Semantic characterization for ASP base revision*

Laurent Garcia[1], Claire Lefevre[1],
Odile Papini[2], Igor Stephan[1], and Éric Würbel[2]

[1] LERIA, Université d'Angers.{garcia,claire.lefevre,stephan}@info.univ-angers.fr
[2] LSIS-CNRS UMR 7296, Aix-Marseille Université.{papini,wurbel}@univ-amu.fr

Abstract. The paper deals with base revision for Answer Set Programming (ASP). Base revision in classical logic stems from the removal of formulas. Exploiting the non-monotonicity of ASP allows one to propose other revision strategies, namely addition strategy or removal and/or addition strategy. These strategies allow one to define families of rule-based revision operators. The paper presents a semantic characterization of these families of revision operators in terms of answer sets. This characterization allows one to equivalently consider the evolution of syntactic logic programs and the evolution of their semantic content.

Keywords: Answer Set Programming, Base revision, Belief revision, Belief change, Non-monotonic reasoning.

1 Introduction

Answer Set Programming (ASP) is an efficient unified formalism for both knowledge representation and reasoning in Artificial Intelligence (AI). It has its roots in non-monotonic reasoning and logic programming and gave rise to intensive research since the Gelfond & Lifschitz's seminal paper [12]. ASP has an elegant and conceptually simple theoretical foundation and has been proved useful for solving a wide range of problems in various domains [26]. Beyond its ability to formalize various problems from AI or to encode combinatorial problems [3, 22], ASP provides also an interesting way to practically solve such problems since some efficient solvers are available [11, 21]. But in most domains, information is subject to change, and so ASP logic programs are subjet to change by addition and/or withdrawal of rules.

Belief change in a classical logic setting, in particular belief revision, has been extensively studied from decades. It applies to situations where an agent faces incomplete or uncertain information and where new and more reliable information may be contradictory with its initial beliefs. Belief revision consists in modifying initial agent's beliefs while taking into account new information and ensuring the consistency of the result. Belief revision relies on three main principles: (i)

* This work was supported by the projects ASPIQ (ANR-12-BS02-0003)

Success: Change must succeed, new information has to be accepted, (*ii*) *Consistency*: The result of the revision operation must be a consistent set of beliefs, and (*iii*) *Minimal change*: The initial beliefs have to be changed as least as possible. Two main frameworks became standards according to the nature of the involved representation of beliefs: AGM paradigm [1] for belief sets revision, rephrased by Katsuno and Mendelzon [18] for model-based revision and Hansson's approach [14] for formula-based revision (or base revision). Several concrete base revision operators have been proposed. Most approaches focus on the construction of consistent subbases maximal w.r.t. several criteria [5, 20]. From a dual point of view, others stem from the minimal withdrawal of formulas in order to restore consistency with new information like Kernel revision [13] or like Removed Sets Revision (RSR) [23, 30, 4] that focuses on subsets of formulas minimal w.r.t. cardinality to remove. All these approaches require selection functions that encode the revision strategies for selecting among subbases or among subsets of formulas to remove.

This paper aims at studying base revision when beliefs are represented by ASP logic programs. Due to the non-monotonic nature of logic programs under answer set semantics, the problem of change in ASP is different and more difficult than in classical logic.

The first approaches dealing with logic programs in a dynamic setting focused on the problem of logic program update [31, 25, 2, 10]. The first work bridging ASP in a dynamic setting and belief change has been proposed by Delgrande and al. [7–9]. Their approach uses a semantic of logic programs in terms of SE-models [29]. Model-based revision and merging stemming from a distance between interpretations have been extended to logic programs. However, they noted that this approach has the drawback that arbitrary sets of SE-models may not necessarily be expressed via a logic program. Recently this drawback has been avoided for classes of logic programs satisfying an AGM-compliance condition on SE-models and a new postulate [6]. Model-based update has been addressed in the same spirit by Slota and Leite [27, 28].

From a syntactic point of view formula-based belief merging and revision have been extented to ASP. The "removed sets" approach for fusion and revision (RSF)[15] and (RSR) [4] respectively, proposed in propositional logic have been extended to ASP [16, 17]. The "remainder sets" approach for screened consolidation in a classical setting has been extended to ASP [19]. The strategy of these two approaches stems from the removal of some rules in order to restore consistency. More recently, a new approach for extending belief base revision to ASP has been proposed with additional strategies stemming from the addition and the addition and/or removal of some rules [33].

This paper focuses on three different families of ASP base revision operators. It first reminds the RSR family, it then introduces the notions of "added Sets" and "modified Sets" and proposes the Added Set Revision (ASR) and the Modified Set Revision (MSR). Note that these families of operators differ from the ones provided in [33] since the minimality criterion for the removed, added or modified sets is cardinality and not set inclusion. For each family of ASP base

revision operators the paper provides a semantic counter-part that characterizes the operators in terms of answer sets.

The main contribution of the paper is the characterization of ASP base revision operators which also covers the family of SLP operators proposed in [33]. This is an important result since it provides a new semantic characterization of logic program revision in terms of answer sets and allows one to change the focus from the evolution of a syntactic logic program to the evolution of its semantic content.

The paper is organized as follows. Section 2 gives a refresher on ASP and on belief base revision. Section 3 recalls RSR revision and provides a semantic characterization of removed sets. Section 4 introduces the notions of added set and ASR revision, it then gives a semantic characterization of added sets. Section 5 introduces the notions of modified set and MSR revision, it then provides a semantic characterization of modified sets. Finally Section 6 concludes the paper.

Due to space limitations, the complete proofs of the theorems are not included, they can be found on `http://aspiq.lsis.org/aspiq/pdf/proofs-2017.pdf`.

2 Preliminaries

In this paper we only consider normal logic programs. Let $\mathcal{A}$ be a set of propositional atoms, a logic program is a finite set of rules of the form:
$(c \leftarrow a_1,\ \ldots,\ a_n,\ not\ b_1,\ \ldots,\ not\ b_m.)$ $n \geq 0, m \geq 0$ where $c, a_1, \ldots, a_n$, $b_1, \ldots b_m \in \mathcal{A}$. The set of all logic programs is denoted by $\mathcal{P}$. The symbol "not" represents default negation and such a program may be seen as a sub-case of default theory of Reiter [24]. A negation-free program is a definite program. For each rule r, let $head(r) = c$, $body^+(r) = \{a_1, \ldots, a_n\}$ and $body^-(r) = \{b_1, \ldots, b_m\}$. If $body^+(r) = \emptyset$ and $body^-(r) = \emptyset$ then the rule is simply written $(c.)$ and is called a fact.

Let X be a set of atoms. A rule r is *applicable in* X if $body^+(r) \subseteq X$. $App(P, X)$ denotes the set of applicable rules of P in X. The least Herbrand model of a definite program P, denoted $Cn(P)$, is the smallest set of atoms closed under P and can be computed as the least fix-point of the following consequence operator: $T_P : 2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$ such that $T_P(X) = Head(App(P, X))$.

The Gelfond-Lifschitz reduct of a program P by a set of atoms X [12] is the program $P^X = \{head(r) \leftarrow body^+(r) \mid r \in P, body^-(r) \cap X = \emptyset\}$. Since it has no default negation, such a program is definite and then it has a unique minimal Herbrand model. By definition, an answer set (or stable model) of P is a set of atoms $X \subseteq \mathcal{A}$ such that $X = Cn(P^X)$. The set of answer sets of a logic program P is denoted by $AS(P)$ and if $AS(P) \neq \emptyset$ the program is said *consistent* otherwise it is said *inconsistent*.

$GR(P, X) = \{r \in P \mid body(r)^+ \subseteq X$ and $body(r)^- \cap X = \emptyset\}$ denotes the set of the generating rules of a logic program P w.r.t. a set of atoms X. A set of rules $R \subseteq P$ is *grounded* if there exists some enumeration $\langle r_i \rangle_{i=1}^n$ of the rules of R such that $\forall i > 0$, $body^+(r_i) \subseteq \{head(r_j) \mid j < i\}$. With those definitions the

following result holds: $X \in AS(P)$ if and only if $X = Cn(GR(P,X)^+)$ if and only if $X = Head(GR(P,X))$ and $GR(P,X)$ is grounded.

A *constraint* is a rule without head ($\leftarrow a_1,\ \ldots,\ a_n,\ not\ b_1,\ \ldots,\ not\ b_m.$) that should be read as ($h \leftarrow a_1,\ \ldots,\ a_n,\ not\ b_1,\ \ldots,\ not\ b_m,\ not\ h.$) where h is a new atom symbol appearing nowhere else in the program.

We now consider a rule ($c \leftarrow a_1,\ \ldots,\ a_n,\ not\ b_1,\ \ldots,\ not\ b_m.$) as a classical implication ($a_1 \wedge \cdots \wedge a_n \wedge \neg b_1 \wedge \cdots \wedge \neg b_m \rightarrow c$) which is equivalent to ($\neg a_1 \vee \cdots \vee \neg a_n \vee b_1 \vee \cdots \vee b_m \vee c$). Hence, an *interpretation* of P is a set of atoms $m \subseteq \mathcal{A}$. An interpretation m *satisfies* a rule r if $body^+(r) \not\subseteq m$ or $body^-(r) \cap m \neq \emptyset$ or $head(r) \in m$. An interpretation m is a *model* of a program P if m satisfies all rules from P. $Mod(P)$ denotes the set of all the models of a logic program P. Conversely, an interpretation m falsifies a rule r if m does not satisfy r: $body^+(r) \subseteq m$ and $body^-(r) \cap m = \emptyset$ and $head(r) \notin m$. $Fal(P,m) = \{r \in P\ :\ body(r)^+ \subseteq m,\ body(r)^- \cap m = \emptyset,\ head(r) \notin m\}$ denotes the set of the rules of a logic program P that are falsified w.r.t. an interpretation m.

The set $m \backslash Cn(GR(P,m)^+)$ denotes the atoms of an interpretation m not deduced from this interpretation for a logic program P and $Nded(m,P) = fact(m \backslash Cn(GR(P,m)^+))$ denotes the previous set of atoms considered as a set of facts.

We remind some notions and notations useful in subsequent sections.

A preorder on a set A is a reflexive and transitive binary relation. A total preorder, denoted by $\leq$, is a preorder such that $\forall x, y \in A$ either $x \leq y$ or $y \leq x$ holds. The equivalence is defined by $x \simeq y$ if and only if $x \leq y$ and $y \leq x$. The corresponding strict total preorder, denoted by $<$, is the relation defined by $x < y$ if and only if $x \leq y$ holds but $x \simeq y$ does not hold. Let M be a subset of A, the set of minimal elements of M with respect to $\leq$, denoted by $Min(M, \leq)$, is defined as: $Min(M, \leq) = \{x \in M, \nexists y \in M : y < x\}$.

Let X and Y be two sets, $|X|$ (resp. $|Y|$) denotes the cardinality of X (resp. of Y) and $X \leq Y$ if $|X| \leq |Y|$. $X \leq Y$ means that X is preferred to Y.

Let A be a finite set, a selection function denoted by f is a function from 2^A to A which for any set $X \in 2^A$ returns an element $f(X)$ such that $f(X) \in X$. If $X = \emptyset$, then the function f is not defined.

3 ASP base revision by removal

Let P and Q be logic programs, revising P by Q is providing a new consistent logic program containing Q and differing as few as possible from $P \cup Q$. This section is dedicated to ASP base revision by removal. This revision strategy stems from the suppression of rules of P when $P \cup Q$ is inconsistent. This strategy is a direct application of the one used for revising belief bases in a classical setting, however it differs from it due to the non-monotonicity of logic programs. For instance, in ASP, P and Q can be inconsistent while $P \cup Q$ is consistent. This is the reason why, we allow P and Q to be inconsistent (note that, in a classical setting, P and Q must be consistent, only $P \cup Q$ may be inconsistent). However, this means that revision is not always possible (for example, when Q does not

admit any classical model since the revision strategy only removes rules from P).

3.1 Rule-based revision by removal

We remind Removed Sets Revision (RSR) extended to ASP [17]. This strategy focuses on the minimal number of rules to remove in order to restore consistency. We first remind the notion of *potential removed set*.

Definition 1 (potential removed set [17]). *Let P, Q be two logic programs, let X be a set of rules. A potential removed set X is such that: (i) $X \subseteq P$. (ii) $(P \setminus X) \cup Q$ is consistent. (iii) For each $X' \subset X$, $(P \setminus X') \cup Q$ is inconsistent.*

$\mathcal{PR}(P, Q)$ denotes the set of potential removed sets for P and Q. According to the definition, if $P \cup Q$ is consistent then $\mathcal{PR}(P, Q) = \{\emptyset\}$. Since potential removed sets are built by removing only rules from P in order to restore consistency of $P \cup Q$, it may be possible that the set of potential removed sets for an inconsistent set Q is empty.

Example 1. Let P and Q be two logic programs such that $P = \{r_1 : a \leftarrow b., r_2 : a., r_3 : b., r_4 : c \leftarrow not\, a., r_5 : d., r_6 : d \leftarrow not\, b.\}$ and $Q = \{\leftarrow a, b., \leftarrow not\, c, d.\}$. These two logic programs are consistent since $AS(P) = \{\{a, b, d\}\}$ and $AS(Q) = \{\emptyset\}$ but $P \cup Q$ is inconsistent. $\mathcal{PR}(P, Q) = \{\{r_3, r_5, r_6\}, \{r_2, r_3\}, \{r_1, r_2\}\}$.

For RSR, the minimality criterion is cardinality, we remind the notion of *removed set* by selecting the potential removed sets minimal w.r.t. cardinality.

Definition 2 (removed set [17]). *Let P, Q be two logic programs, let X be a set of rules. A removed set X is such that: (i) X is a potential removed set. (ii) There is no potential removed set Y such that $Y < X$.*

$\mathcal{R}(P, Q)$ denotes the set of removed sets for P and Q. According to the definition $\mathcal{R}(P, Q) = Min(\mathcal{PR}(P, Q), \leq)$ and if $P \cup Q$ is consistent then $\mathcal{R}(P, Q) = \{\emptyset\}$.

Example 2 (Example 1 continued). $\mathcal{R}(P, Q) = \{\{r_2, r_3\}, \{r_1, r_2\}\}$.

We now remind the *Removed Set Revision* (RSR) family of operators.

Definition 3 (RSR operators [17]). *Let P, Q be two logic programs, $\mathcal{R}(P, Q)$ be the set of removed sets and f be a selection function. The revision operator denoted by $\star_{RSR(f)}$ is a function from $\mathcal{P} \times \mathcal{P}$ to $\mathcal{P}$ such that $P \star_{RSR(f)} Q = (P \setminus f(\mathcal{R}(P, Q))) \cup Q$.*

Note that if $\mathcal{R}(P, Q) = \emptyset$, $f(\mathcal{R}(P, Q))$ is not defined. This means that the program P cannot be revised by Q.

3.2 Semantic characterization of ASP base revision by removal

We now present the semantical counterparts of the potential removed set and removed set notions. We first introduce the notion of *canonical removed set*. Intuitively, given P and Q two logic programs, a *canonical removed set* is a set of rules of P falsified by a model of Q.

Definition 4 (canonical removed set). *Let P, Q be two logic programs and m be a model of Q. A* canonical removed set *X is such that: (i) $X = Fal(P, m)$. (ii) $m \in AS((P \setminus X) \cup Q)$.*

$CR(P, Q, m) = \{X \mid X = Fal(P, m)$ and $m \in AS((P \setminus X) \cup Q)\}$ denotes the set of all canonical removed sets for m and $CR(P, Q) = \bigcup_{m \in Mod(Q)} CR(P, Q, m)$ denotes the union set of all canonical removed sets for the models of a program Q w.r.t. a program P. Note that for one given interpretation m, there is zero or one canonical removed set and if Q has no model then $CR(P, Q) = \emptyset$.

Example 3 (Example 1 continued). $P = \{r_1 : a \leftarrow b., r_2 : a., r_3 : b., r_4 : c \leftarrow not\ a., r_5 : d., r_6 : d \leftarrow not\ b.\}$ and $Q = \{\leftarrow a, b., \leftarrow not\ c, d.\}$.

$m \in Mod(Q)$	$X = Fal(P, m)$	$AS((P \setminus X) \cup Q)$	$CR(P, Q, m)$
$\emptyset$	$\{r_2, r_3, r_4, r_5, r_6\}$	$\{\emptyset\}$	$\{Fal(P, \emptyset)\}$
$\{a\}$	$\{r_3, r_5, r_6\}$	$\{\{a\}\}$	$\{Fal(P, \{a\})\}$
$\{b\}$	$\{r_1, r_2, r_4, r_5\}$	$\{\{b\}\}$	$\{Fal(P, \{b\})\}$
$\{c, d\}$	$\{r_2, r_3\}$	$\{\{c, d\}\}$	$\{Fal(P, \{c, d\})\}$
$\{a, c, d\}$	$\{r_3\}$	$\emptyset$	$\emptyset$
$\{b, c, d\}$	$\{r_1, r_2\}$	$\{\{b, c, d\}\}$	$\{Fal(P, \{b, c, d\})\}$
$\{a, c\}$	$\{r_3, r_5, r_6\}$	$\{\{a\}\}$	$\emptyset$
$\{b, c\}$	$\{r_1, r_2, r_5\}$	$\{\{b, c\}\}$	$\{Fal(P, \{b, c\})\}$
$\{c\}$	$\{r_2, r_3, r_5, r_6\}$	$\{\{c\}\}$	$\{Fal(P, \{c\})\}$

The last column of the table gives the set of canonical removed sets corresponding to a classical model of Q given in the first column of the table. Hence, if we restrict our attention to minimal canonical removed sets w.r.t. inclusion, we have $Min(CR(P, Q), \subseteq) = \{\{r_2, r_3\}, \{r_1, r_2\}, \{r_2, r_3, r_6\}\}$ and, if we consider minimality w.r.t. cardinality, we have $Min(CR(P, Q), \leq) = \{\{r_2, r_3\}, \{r_1, r_2\}\}$.

The following theorems give the equivalence between syntactic (potential) removed sets and semantic canonical removed sets.

Theorem 1. *Let P and Q be logic programs. $\mathcal{PR}(P, Q) = Min(CR(P, Q), \subseteq)$.*

Proof. (sketch) The proof is based on the fact that the rules of a potential removed set X are exactly the rules of P falsified by the answer sets of $(P \setminus X) \cup Q$.

The following theorem is a direct consequence of Definition 2 and Theorem 1.

Theorem 2. *Let P and Q be logic programs. $\mathcal{R}(P, Q) = Min(CR(P, Q), \leq)$.*

We introduce a preference relation between interpretations, denoted by $<_{R(P)}$ as follows. Let m and m' be two interpretations and P be a logic program, $m <_{R(P)} m'$ means that $|Fal(P,m)| < |Fal(P,m')|$.

The following result directly follows from Theorem 2 and Definition 3. It provides a semantic characterization of RSR operators for logic programs.

Theorem 3. *Let P and Q be logic programs.*
Let $M = \{m \in Mod(Q)$ s.t. $CR(P,Q,m) \neq \emptyset\}$. (i) For each selection function f, if $m \in AS(P \star_{RSR(f)} Q)$ then $m \in Min(M, \leq_{R(P)})$. (ii) If $m \in Min(M, \leq_{R(P)})$ then there exists a selection function f s.t. $m \in AS(P \star_{RSR(f)} Q)$.

Example 4 (Example 1 continued). $P = \{r_1 : a \leftarrow b., r_2 : a., r_3 : b., r_4 : c \leftarrow not\, a., r_5 : d., r_6 : d \leftarrow not\, b.\}$ and $Q = \{\leftarrow a, b., \leftarrow not\, c, d.\}$. From the table in Example 3 we have $\mathcal{PR}(P,Q) = Min(CR(P,Q), \subseteq)$, $\mathcal{R}(P,Q) = Min(CR(P,Q), \leq)$ and $Min(M, \leq_{R(P)}) = \{\{c,d\}, \{b,c,d\}\}$. Let f_1 and f_2 be the functions that select respectively $\{r_2, r_3\}$ and $\{r_1, r_2\}$ the respective revised logic programs are $P \star_{RSR(f_1)} Q = P \backslash \{r_2, r_3\} \cup Q$ and $P \star_{RSR(f_2)} Q = P \backslash \{r_1, r_2\} \cup Q$ with $AS(P \star_{RSR(f_1)} Q) = \{\{c,d\}\}$ and $AS(P \star_{RSR(f_2)} Q) = \{\{b,c,d\}\}$.

4 ASP base revision by addition

This section is dedicated to ASP base revision by addition. Let P and Q be logic programs, this revision strategy stems from the addition of rules to P when $P \cup Q$ is inconsistent. This strategy relies on the non-monotonicity of the ASP framework. Indeed, adding a new rule may block a rule which contributes to inconsistency. We allow P and Q to be inconsistent. Note that, with this strategy, revision is not always possible, even if P and Q are consistent. Revising by addition allows for adding any kind of rules but adding rules is equivalent to adding a set of facts which makes the revision process easier.

4.1 Rule-based revision by addition

The strategy of Added Set Revision (ASR) focuses on the minimal number of new rules to add in order to restore consistency. We first introduce the notion of *potential added set*.

Definition 5 (potential added set). *Let P and Q be two logic programs, let Y be a set of rules. A potential added set Y is such that: (i) $(P \cup Y) \cup Q$ is consistent. (ii) For each $Y' \subset Y$, $(P \cup Y') \cup Q$ is inconsistent.*

$\mathcal{PA}(P,Q)$ denotes the set of potential added sets for P and Q. Note that rules from a potential added set Y cannot already belong to $P \cup Q$. Indeed, if $(P \cup Y) \cup Q$ is consistent and some rule r from Y already belongs to $P \cup Q$, then there exists some $Y' = Y \backslash \{r\}$ such that $Y' \subseteq Y$ and $(P \cup Y') \cup Q = (P \cup Y) \cup Q$ is consistent. According to the definition if $P \cup Q$ is consistent then $\mathcal{PA}(P,Q) = \{\emptyset\}$.

Example 5. Let P and Q be two logic programs such that $P = \{a \leftarrow not\ b.\}$ and $Q = \{\leftarrow a, not\ c., \leftarrow a, not\ d.\}$. If we restrict ourselves to the addition of facts, we have $\mathcal{PA}(P,Q) = \{\{c.,\ d.\}, \{b.\}\}$.

Note that revision by addition is not always feasible, even if P and Q are consistent.

Example 6. Let P and Q be two logic programs such that $P = \{a.\}$ and $Q = \{\leftarrow a.\}$. We have $\mathcal{PA}(P,Q) = \emptyset$. This is because, even if the constraint $\leftarrow a.$ can be read as $h \leftarrow a, not\ h.$, we do not allow to use the implicit atom h for adding rules and thus the constraint $\leftarrow a.$ cannot be blocked by addition.

For ASR, the minimality criterion is cardinality, we introduce the notion of *added set* by selecting the potential added sets minimal w.r.t. cardinality.

Definition 6 (added set). *Let P and Q be two logic programs, let Y be a set of rules. An added set Y is such that: (i) Y is a potential added set. (ii) There is no potential added set Z such that $Z < Y$.*

$\mathcal{A}(P,Q)$ denotes the set of added sets for P and Q. According to the definition $\mathcal{A}(P,Q) = Min(\mathcal{PA}(P,Q), \leq)$ and if $P \cup Q$ is consistent then $\mathcal{A}(P,Q) = \{\emptyset\}$.

Example 7 (Example 5 continued). For these programs, $\mathcal{A}(P,Q) = \{\{b.\}\}$ is reduced to only one added set.

Example 8. Let P and Q be two logic programs such that $P = \{a \leftarrow b., b \leftarrow a., c.\}$ and $Q = \{\leftarrow c, not\ a.\}$. We have $\mathcal{PA}(P,Q) = \mathcal{A}(P,Q) = \{\{a.\}, \{b.\}\}$.

We now define the *Added Set Revision* family of operators.

Definition 7 (ASR operators). *Let P, Q be two logic programs, $\mathcal{A}(P,Q)$ be the set of added sets and f be a selection function. The revision operator denoted by $\star_{ASR(f)}$ is a function from $\mathcal{P} \times \mathcal{P}$ to $\mathcal{P}$ such that $P \star_{ASR(f)} Q = (P \cup f(\mathcal{A}(P,Q))) \cup Q$.*

Note that if $\mathcal{A}(P,Q) = \emptyset$, $f(\mathcal{A}(P,Q))$ is not defined. That means that the program cannot be revised by addition.

4.2 Semantic characterization of ASP base revision by addition

We now present the semantic counterparts of the potential added set and added set notions. We first introduce the notion of *canonical added set*. Given P and Q two logic programs, a *canonical added set* is a set of facts corresponding to the least subset (w.r.t. inclusion) of atoms to add to $P \cup Q$ so that a model of $P \cup Q$ becomes an answer set.

Definition 8 (canonical added set). *Let P and Q be two logic programs and m be a model of $P \cup Q$. A* canonical added set *Y is such that: (i) $Y \subseteq Nded(m, P \cup Q)$. (ii) $m \in AS(P \cup Q \cup Y)$. (iii) $\forall Y' \subset Y$, $m \notin AS(P \cup Q \cup Y')$.*

$CA(P,Q,m)$ denotes the set of all canonical added sets for m and $CA(P,Q) = \bigcup_{m \in Mod(P \cup Q)} CA(P,Q,m)$. Note that $CA(P,Q,m) = Min(\{Y \mid Y \subseteq Nded(m, P \cup Q)$ and $m \in AS(P \cup Q \cup Y)\}, \subseteq)$.

Example 9. Let $P = \{r_1 : a \leftarrow b, not\ c., r_2 : c \leftarrow d, e, not\ a., r_3 : b \leftarrow d., r_4 : d \leftarrow b., r_5 : e.\}$ and $Q = \{r_6 :\leftarrow not\ a, not\ c., r_7 :\leftarrow a, not\, b, not\ c., r_8 :\leftarrow c, not\, d, not\ a.\}$.

$m \in Mod(P \cup Q)$	$GR(P \cup Q, m)$	$Nded(m, P \cup Q)$	Y	$AS(P \cup Q \cup Y)$	$CA(P,Q,m)$
$\{a,c,e\}$	$\{r_5\}$	$\{a.,c.\}$	$\{a.,c.\}$	$\{\{a,c,e\}\}$	$\{\{a.,c.\}\}$
$\{a,b,d,e\}$	$\{r_1,r_3,r_4,r_5\}$	$\{a.,b.,d.\}$	$\{b.\}$	$\{\{a,b,d,e\}, \{b,c,d,e\}\}$	
			$\{d.\}$	$\{\{a,b,d,e\}, \{b,c,d,e\}\}$	$\{\{b.\},\{d.\}\}$
$\{b,c,d,e\}$	$\{r_2,r_3,r_4,r_5\}$	$\{b.,c.,d.\}$	$\{b.\}$	$\{\{a,b,d,e\}, \{b,c,d,e\}\}$	
			$\{d.\}$	$\{\{a,b,d,e\}, \{b,c,d,e\}\}$	$\{\{b.\},\{d.\}\}$
$\{a,b,c,d,e\}$	$\{r_3,r_4,r_5\}$	$\{a.,b.,c.,d.\}$	$\{a.,b.,c.\}$	$\{\{a,b,c,d,e\}\}$	$\{\{a.,b.,c.\},$
			$\{a.,c.,d.\}$	$\{\{a,b,c,d,e\}\}$	$\{a.,c.,d.\}\}$

The last column of the table gives the set of canonical added sets corresponding to a classical model of $P \cup Q$ given in the first column of the table. Hence $Min(CA(P,Q), \subseteq) = \{\{a.,c.\},\{b.\},\{d.\}\}$ and $Min(CA(P,Q), \leq) = \{\{b.\},\{d.\}\}$.

Note that canonical added sets only consists of facts. Thus the semantic characterization of ASR operators is limited to ASR operators that require the addition of facts (not the addition of general rules).

Theorem 4. *Let P, Q be logic programs. $\mathcal{PA}(P,Q) = Min(CA(P,Q), \subseteq)$.*

Proof. (sketch) The proof is based on the fact that the rules (facts) of a potential added set Y are a subset of the (facts corresponding to the) atoms from the answer sets of $P \cup Y \cup Q$ that can not be deduced from $P \cup Q$.

The following theorem is a direct consequence of Theorem 4 and Definition 6.

Theorem 5. *Let P, Q be logic programs. $\mathcal{A}(P,Q) = Min(CA(P,Q), \leq)$.*

We introduce a preference relation between interpretations, denoted by $<_{A(P,Q)}$ as follows. Let m and m' are two interpretations, $m <_{A(P,Q)} m'$ means that $Min(CA(P,Q,m), \leq) < Min(CA(P,Q,m'), \leq)$.

The following result directly follows from Theorem 5 and Definition 7. It provides a semantic characterization of ASR revision operators for logic programs.

Theorem 6. *Let P, Q be two logic programs.*
Let $M = \{m \in Mod(Q)$ s.t. $CA(P,Q,m) \neq \emptyset\}$. (i) For each f, if $m \in AS(P \star_{ASR(f)} Q)$ then $m \in Min(M, \leq_{A(P,Q)})$. (ii) If $m \in Min(M, \leq_{A(P,Q)})$ then there exists f s.t. $m \in AS(P \star_{ASR(f)} Q)$.

Example 10 (Example 9 continued). Let $P = \{a \leftarrow b\ , not\ c., c \leftarrow d\ , e\ , not\ a., b \leftarrow d., d \leftarrow b., e.\}$ and $Q = \{\leftarrow not\ a\ , not\ c., \leftarrow a, not\ b\ , not\ c., \leftarrow c, not\ d\ , not\ a.\}$. From the table in Example 9 we have $\mathcal{PA}(P,Q) = Min(CA(P,Q), \subseteq)$, $\mathcal{A}(P,Q) = Min(CA(P,Q), \leq) = \{\{b.\}, \{d.\}\}$. $Min(M, \leq_{A(P,Q)}) = \{\{a,b,d,e\}, \{b,c,d,e\}\}$. Let f_1 and f_2 be the functions that select respectively $\{b.\}$ and $\{d.\}$ the respective revised logic programs are $P \star_{ASR(f_1)} Q = P \cup Q \cup \{b.\}$ and $P \star_{ASR(f_2)} Q = P \cup Q \cup \{d.\}$ with $AS(P \star_{ASR(f_1)} Q) = AS(P \star_{ASR(f_2)} Q) = \{\{a,b,d,e\}, \{b,c,d,e\}\}$.

5 ASP base revision by modification

This section now focuses on ASP base revision by modification. Modification strategy means combining the removal strategy and the addition one. Let P and Q be logic programs, removing some rules from P and in the same time adding some new rules to P allows one to construct a new logic program which is consistent with Q and differs the least from P. Indeed, revision by removal and revision by addition can be viewed as particular cases of revision by modification.

5.1 Rule-based revision by modification

The strategy of Modified Set Revision (MSR) focuses on the minimal number of rules to remove and/or to add in order to restore consistency. A (potential) modified set is a pair of sets of rules, where the first component is the set of rules to remove and the second one is the set of new rules to add. We first define preference relations between pairs of sets of rules w.r.t. set inclusion and w.r.t. cardinality as follows.

Definition 9. *Let X, Y, X', Y' be sets of rules. $(X',Y') \subset (X,Y)$ if $X' \subset X$ and $Y' \subseteq Y$, or $X' \subseteq X$ and $Y' \subset Y$. $(X',Y') \leq (X,Y)$ if $|X' \cup Y'| \leq |X \cup Y|$.*

We now introduce the notion of *potential modified set*.

Definition 10 (potential modified set). *Let P, Q be two logic programs, let (X,Y) be a pair of sets of rules. A potential modified set (X,Y) is such that: (i) $X \subseteq P$. (ii) $(P \backslash X) \cup Y \cup Q$ is consistent. (iii) For each (X',Y') such that $(X',Y') \subset (X,Y)$, $(P \backslash X') \cup Y' \cup Q$ is inconsistent.*

$\mathcal{PM}(P,Q)$ denotes the set of potential modified sets for P and Q. According to the definition if $P \cup Q$ is consistent then $\mathcal{PM}(P,Q) = \{(\emptyset, \emptyset)\}$.

Example 11. Let $P = \{r_1 : a \leftarrow not\ b.,\ r_2 : c \leftarrow not\ b.,\ r_3 :\leftarrow f.\}$ and $Q = \{\leftarrow a.,\ \leftarrow c.,\ f \leftarrow not\ g.,\ f \leftarrow not\ h.\}$. $\mathcal{PM}(P,Q) = \{(\{r_1, r_2, r_3\}, \emptyset),\ (\emptyset, \{b., g., h.\}),\ (\{r_3\}, \{b.\}),\ (\{r_1, r_2\},\ \{g., h.\})\}$.

As for RSR and ASR, the minimality criterion for MSR is cardinality, we introduce the notion of *modified set* by selecting the potential modified sets minimal w.r.t. cardinality.

Definition 11 (modified set). *Let P, Q be two logic programs, let X and Y be two sets of rules. A modified set (X, Y) is such that: (i) (X, Y) is a potential modified set. (ii) There is no potential modified set (X', Y') such that $(X', Y') < (X, Y)$.*

We denote by $\mathcal{M}(P, Q)$ the set of modified sets. According to the definition $\mathcal{M}(P, Q) = Min(\mathcal{PM}(P, Q), \leq)$ and if $P \cup Q$ is consistent then $\mathcal{M}(P, Q) = \{(\emptyset, \emptyset)\}$.

Example 12 (Example 11 continued). $\mathcal{M}(P, Q) = \{(\{r_3\}, \{b.\})\}$.

We now define the *Modified Set Revision* family of operators.

Definition 12 (MSR operators). *Let P, Q be two logic programs, $\mathcal{M}(P, Q)$ be the set of modified sets and f be a selection function. The revision operator denoted by $\star_{MSR(f)}$ is a function from $\mathcal{P} \times \mathcal{P}$ to $\mathcal{P}$ such that $P \star_{MSR(f)} Q = (P \setminus X) \cup Y \cup Q$ where $(X, Y) = f(\mathcal{M}(P, Q))$.*

5.2 Semantic characterization of ASP base revision by modification

We now present the semantic counterpart of potential modified set and modified set notions. We first introduce the notion of *canonical modified set*. Given P and Q two logic programs, a canonical modified set is a pair of sets (X, Y) where X is the set of rules from P falsified by a model of Q and Y is the set of facts corresponding to least subsets of atoms of a model of Q not deduced from $(P \backslash X) \cup Q$.

Definition 13 (canonical modified set). *Let P, Q be two logic programs and m be a model of Q. A* canonical modified set *(X, Y) is such that: (i) $X = Fal(P, m)$ (ii) $Y \subseteq Nded(m, (P \setminus X) \cup Q)$ (iii) $m \in AS((P \setminus X) \cup Q \cup Y)$ (iv) $\forall (X', Y') \subset (X, Y)$, $m \notin AS((P \setminus X') \cup Q \cup Y')$.*

$CM(P, Q, m) = Min(\{(X, Y) \mid X = Fal(P, m), Y \subseteq Nded(m, (P \backslash X) \cup Q)$ and $m \in AS((P \setminus X) \cup Q \cup Y)\}, \subseteq)$ denotes the set of all canonical modified sets for m and $CM(P, Q) = \bigcup_{m \in Mod(Q)} CM(P, Q, m)$.

Example 13 (Example 11 continued). $P = \{r_1 : a \leftarrow not\ b.,\ r_2 : c \leftarrow not\ b.,\ r_3 : \leftarrow f.\}$ and $Q = \{\leftarrow a.,\ \leftarrow c.,\ f \leftarrow not\ g.,\ f \leftarrow not\ h.\}$.

$m \in Mod(Q)$	$X = Fal(P, m)$	$Y \subseteq Nded((P \setminus X) \cup Q, m)$	$AS((P \setminus X) \cup Q \cup Y)$
$\{f\}$	$\{r_1, r_2, r_3\}$	$\emptyset$	$\{\{f\}\}$
$\{b, f\}$	$\{r_3\}$	$\{b.\}$	$\{\{b, f\}\}$
$\{f, g\}$	$\{r_1, r_2, r_3\}$	$\{g.\}$	$\{\{f, g\}\}$
$\{f, h\}$	$\{r_1, r_2, r_3\}$	$\{h.\}$	$\{\{f, h\}\}$
$\{b, f, g\}$	$\{r_3\}$	$\{b., g.\}$	$\{\{b, f, g\}\}$
$\{b, f, h\}$	$\{r_3\}$	$\{b., h.\}$	$\{\{b, f, h\}\}$
$\{g, h\}$	$\{r_1, r_2\}$	$\{g., h.\}$	$\{\{g, h\}\}$
$\{f, g, h\}$	$\{r_1, r_2, r_3\}$	$\{f., g., h.\}$	$\{\{f, g, h\}\}$
$\{b, g, h\}$	$\emptyset$	$\{b., g., h.\}$	$\{\{b, g, h\}\}$
$\{b, f, g, h\}$	$\emptyset$	$\{b., f., g., h.\}$	$\{\{b, f, g, h\}\}$

The second and the third column of the table give the first and the second component respectively of the canonical modified set corresponding to a classical model of Q given in the first column of the table. Hence $Min(CM(P,Q),\subseteq) = \{(\{r_1,r_2,r_3\},\emptyset),\ (\{r_3\},\{b.\}),\ (\{r_1,r_2\},\{g.,h.\}),\ (\emptyset,\{b.,g.,h.\})\}$ and $Min(CM(P,Q),\leq) = \{(\{r_3\},\{b.\})\}$.

Theorem 7. *Let P, Q be programs. $\mathcal{PM}(P,Q) = Min(CM(P,Q),\subseteq)$.*

Proof. (sketch) The proof is based on the fact that if (X,Y) is a potential modified set of P and Q, then X is a potential removed set of P and $Q \cup Y$, and Y is a potential added set of $P \setminus X$ and Q.

The following theorem is a direct consequence of Theorem 7 and Definition 11.

Theorem 8. *Let P, Q be logic programs. $\mathcal{M}(P,Q) = Min(CM(P,Q),\leq)$.*

We introduce a preference relation between interpretations, denoted by $<_{M(P,Q)}$. Let m and m' be interpretations, $m <_{M(P,Q)} m'$ means that $Min(CM(P,Q,m),\leq) < Min(CM(P,Q,m'),\leq)$. The following result directly follows from Theorem 8 and Definition 12. It provides a semantic characterization of MSR revision operators.

Theorem 9. *Let P, Q be logic programs.*
Let $M = \{m \in Mod(Q)$ s.t. $CM(P,Q,m) \neq \emptyset\}$. (i) For each f, if $m \in AS(P \star_{MSR(f)} P)$ then $m \in Min(M, \leq_{M(P,Q)})$ (ii) If $m \in Min(M, \leq_{M(P,Q)})$ then there exists f s.t. $m \in AS(P \star_{MSR(f)} P)$.

Example 14 (Example 11 continued). $P = \{r_1 : a \leftarrow not\ b.,\ r_2 : c \leftarrow not\ b.,\ r_3 : \leftarrow f.\}$ and $Q = \{\leftarrow a.,\ \leftarrow c.,\ f \leftarrow not\ g.,\ f \leftarrow not\ h.\}$. From the table in Example 13 we have $\mathcal{PM}(P,Q) = Min(CM(P,Q),\subseteq)$, $\mathcal{M}(P,Q) = Min(CM(P,Q),\leq) = \{(\{r_3\},\{b.\})\}$ and $M = \{\{b,f\}\}$. There is only one modified set thus f selects $(\{r_3\},\{b.\})$ and $P \star_{MSR(f)} Q = \{r_1,\ r_2\} \cup \{b.\} \cup Q$ and $AS(P \star_{MSR(f)} Q) = \{\{b,f\}\}$.

6 Concluding discussion

Belief base revision has first been extended to ASP in [19] with the "remainder Sets" approach and in [16, 17] with the "removed Sets" approach. These two approaches rely on the removal of rules. More recently, another approach called SLP revision [32, 33] has been proposed. The strategy stems from the removal or the addition and/or removal of rules. Let P be the initial logic program and Q be the new one. The removal (respectively addition and addition and/or removal) strategies stem from the construction of "s-removal" (respectively "s-expansion" and "s-compatible") logic programs which are subsets of P consistent with Q maximal w.r.t. set inclusion (respectively sets of rules containing P consistent with Q and minimal w.r.t. set inclusion, and a combination of both for the third strategy). They are the dual sets of potential removed sets, potential added sets and potential modified sets respectively.

Note that the families of revision operators proposed in this paper differ from SLP revision operators since the maximality criterion is set inclusion for SLP whereas the minimality criterion for RSR, ASR and MSR revision operators is cardinality.

Moreover, in this paper we go a step further since we provide a semantic characterization in terms of answer sets for RSR, ASR and MSR revision families of operators. This is an important contribution since it allows one to go from the evolution of a syntactic rule-based revision operator to the evolution of its semantic content.

Future work will be dedicated to the study of logical properties of the proposed revision operators in terms of satisfaction of Hansson's postulates for base revision adapted to the ASP framework. We also plan to implement the proposed families of revision operators and to conduct an experimental study. Another issue is the study of their computational complexity.

References

1. Alchourrón, C.E., Makinson, D.: On the logic of theory change : Safe contraction. Stud. Log. 44(4), 14–37 (1985)
2. Alferes, J.J., Leite, J.A., Pereira, L.M., Przymusinska, H., Przymusinski, T.C.: Dynamic updates of non-monotonic knowledge bases. J. Log. Prog. 45(1-3), 43–70 (2000)
3. Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press (2003)
4. Benferhat, S., Bennaim, J., Papini, O., Würbel, E.: Answer set programming encoding of prioritized removed sets revision: Application to gis. Appl. Intell. 32, 60–87 (2010)
5. Benferhat, S., Cayrol, C., Dubois, D., Lang, J., Prade, H.: Inconsistency management and prioritized syntax-based entailment. In: Proc. of IJCAI'93. pp. 640–645 (1993)
6. Delgrande, J.P., Peppas, P., Woltran, S.: Agm-style belief revision of logic programs under answer set semantics. In: Proc. of LPNMR'13. pp. 264–276 (2013)
7. Delgrande, J.P., Schaub, T., Tompits, H., Woltran, S.: Belief revision of logic programs under answer set semantics. In: Proc. of KR'08. pp. 411–421 (2008)
8. Delgrande, J.P., Schaub, T., Tompits, H., Woltran, S.: Merging logic programs under answer set semantics. In: Proc. of ICLP'09. pp. 160–174 (2009)
9. Delgrande, J.P., Schaub, T., Tompits, H., Woltran, S.: A model-theoretic approach to belief change in answer set programming. ACM Trans. Comput. Log. 14(2), 14:1–14:46 (2013)
10. Eiter, T., Fink, M., Sabbatini, G., Tompits, H.: On properties of update sequences based on causal rejection. TPLP 2(6), 711–767 (2002)
11. Gebser, M., Kaufmann, B., Schaub, T.: Conflict-driven answer set solving: From theory to practice. Art. Intell. 187, 52–89 (2012)
12. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Proc. of ICLP'88. pp. 1070–1080 (1988)
13. Hansson, S.O.: Semi-revision. J.Appl. Non-class. Log. 7, 151–175 (1997)
14. Hansson, S.O.: A text of belief dynamics theory change and database updating (1999)

15. Hué, J., Papini, O., Würbel, E.: Removed sets fusion: Performing off the shelf. In: Proc. of ECAI'08. pp. 94–98 (2008)
16. Hué, J., Papini, O., Würbel, E.: Merging belief bases represented by logic programs. In: Proc. of ECSQARU'09. pp. 371–382 (2009)
17. Hué, J., Papini, O., Würbel, E.: Extending belief base change to logic programs with asp. In: Trends in Belief Revision and Argumentation Dynamics. Studies in Logic (december 2013)
18. Katsuno, H., Mendelzon, A.O.: Propositional knowledge base revision and minimal change. Art. Intell. 52(3), 263–294 (1991)
19. Krümpelmann, P., Kern-Isberner, G.: Belief base change operations for answer set programming. In: Proc. of JELIA'12. pp. 294–306 (2012)
20. Lehmann, D.: Belief revision, revised. In: Proc. of IJCAI'95. pp. 1534–1540 (1995)
21. 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. Log. 7(3), 499–562 (2006)
22. Niemelä, I.: Logic programs with stable model semantics as a constraint programming paradigm. AMAI 25(3-4), 241–273 (1999)
23. Papini, O.: A complete revision function in propositionnal calculus. In: Neumann, B. (ed.) Proc. of ECAI'92. pp. 339–343. John Wiley and Sons. Ltd (1992)
24. Reiter, R.: A logic for default reasoning. Art. Intell. 13(1-2), 81–132 (1980)
25. Sakama, C., Inoue, K.: Updating extended logic programs through abduction. In: Proc. of LPNMR'99. pp. 147–161 (1999)
26. Schaub, T.: Here's the beef: Answer set programming ! In: Proc. of ICLP'08. pp. 93–98 (2008)
27. Slota, M., Leite, J.: On semantic update operators for answer-set programs. In: Proc. of ECAI'10. pp. 957–962 (2010)
28. Slota, M., Leite, J.: The rise and fall of semantic rule updates based on se-models. TPLP 14(6), 869–907 (2014)
29. Turner, H.: Strong equivalence made easy: nested expressions and weight constraints. TPLP 3, 609–622 (2003)
30. Würbel, E., Jeansoulin, R., O.Papini: Revision : An application in the framework of GIS. In: Proc. of KR'00. pp. 505–518 (2000)
31. Zhang, Y., Foo, N.Y.: Updating logic programs. In: Proc. of ECAI'98. pp. 403–407 (1998)
32. Zhuang, Z., Delgrande, J.P., Nayak, A.C., Sattar, A.: A new approach for revising logic programs. In: Proc. of NMR'16. pp. 171–176 (2016)
33. Zhuang, Z., Delgrande, J.P., Nayak, A.C., Sattar, A.: Reconsidering agm-style belief revision in the context of logic programs. In: Proc. of ECAI'16. pp. 671–679 (2016)