Possibilistic ASP Base Revision by Certain Input

Laurent Garcia[1], **Claire Lefèvre**[1], **Odile Papini**[2], **Igor Stéphan**[1] and **Éric Würbel**[2]
[1] LERIA, Université d'Angers, France
[2] Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
{laurent.garcia,claire.lefevre,igor.stephan}@univ-angers.fr, {odile.papini,eric.wurbel}@univ-amu.fr

Abstract

Belief base revision has been studied within the answer set programming framework. We go a step further by introducing uncertainty and studying belief base revision when beliefs are represented by possibilistic logic programs under possibilistic answer set semantics and revised by certain input. The paper proposes two approaches of rule-based revision operators and presents their semantic characterization in terms of possibilistic distribution. This semantic characterization allows for equivalently considering the evolution of syntactic logic programs and the evolution of their semantic content. It then studies the logical properties of the proposed operators and gives complexity results.

1 Introduction

The paper deals with belief revision within the context of incomplete and uncertain information. Our main motivation comes from reasoning with data from the Web which are commonly represented by means of ontologies and description logics (DL). Such data are often incomplete, imprecise and, supplied by several sources, have different confidence levels. A non-monotonic logic framework able to handle uncertainty is then suitable to address knowledge change in this context.

Belief revision consists in incorporating new information, changing as little as possible an agent's initial beliefs while preserving consistency. After the seminal work of AGM [Alchourrón and Makinson, 1985], belief revision has been extensively studied within a classical setting according to model-based [Katsuno and Mendelzon, 1991] and formula-based (or base) [Hansson, 1999] standpoints.

Little research investigated belief revision when the underlying logic is non-monotonic. Most approaches were proposed for Answer Set Programming (ASP) [Gelfond and Lifschitz, 1988; Schaub, 2008] which is an efficient unified formalism for both knowledge representation and reasoning in Artificial Intelligence. The two main approaches are model-based revision for logic programs [Delgrande *et al.*, 2013; Slota and Leite, 2014], stemming from SE-models [Turner, 2003], and rule-based revision from two standpoints: "remainder sets" [Krümpelmann and Kern-Isberner, 2012] and "removed sets" [Hué *et al.*, 2013] approaches. Rule-based revision relies on the removal of some rules in order to restore consistency, while new strategies stemming from the addition and/or removal of some rules have been proposed [Zhuang *et al.*, 2016; Garcia *et al.*, 2017].

In order to deal with uncertainty in a non-monotonic formalism, ASP was extended by integrating a theory that copes with uncertainty. Probabilities are the widely used one and their applications to ASP has been proposed [Baral *et al.*, 2009; Wang and Lee, 2015; de Morais and Finger, 2013]. Nevertheless, when dealing with knowledge where uncertainty is expressed by subjective levels, a qualitative setting is more suitable and we choose to use possibilities where the values represent a level in a scale [Dubois and Prade, 2001].

ASP was extended to possibilistic ASP [Nicolas *et al.*, 2006; Bauters *et al.*, 2015] with a possibilistic answer set semantics stemming from possibility theory [Dubois and Prade, 1998; 1988]. The initial work [Nicolas *et al.*, 2005] is directly inspired from the classical setting and the answer sets of a possibilistic ASP program match with the (classical) answer sets of the corresponding program without weights. On the other hand, in [Bauters *et al.*, 2010], the strength of the negative body of a rule is taken into account but it requires an additional guess to determine the weights of the answers, thus the resulting answer sets do not match with the classical ones. In the present work, we use the framework proposed in [Nicolas *et al.*, 2006].

Within the context of reasoning with Web applications, there is an increasing interest in bridging ontologies languages and logic programming [Eiter *et al.*, 2008; Motik and Rosati, 2010; Lopes *et al.*, 2017]. In particular, ASP has been extended with existential variables in order to represent DL knowledge bases (KB) [Baget *et al.*, 2018]. Moreover, in a dynamic setting, ASP has been used to repair DL ontologies [Baget *et al.*, 2016]. Besides, uncertain pieces of information can be represented in possibilistic DL-Lite [Benferhat and Bouraoui, 2017]. A step further is to study the dynamics of possibilistic ASP in order to provide a computational framework for the dynamics of possibilistic DL KB. The contribution of this paper is a study of belief base revision within the framework of possibilistic ASP when the input is certain. The certainty of input corresponds to one of the revision principles which specifies that new information has to be accepted.

The paper is organized as follows. After a refresher on possibilistic ASP, two approaches of possibilistic ASP base revision are proposed. The first one, called "standard possibilistic ASP base revision" stems from the strategy of keeping the most certain possibilistic rules beyond the inconsistency degree. However, if the goal is to change as little as possible the original beliefs, it suffers from the "drowning problem" (unnecessary removal of rules not involved in inconsistency). In order to overcome this problem we introduce another approach, called "RSR possibilistic ASP base revision" which is based on the strategy of removing a minimum number of the least certain possibilistic rules, keeping the possibilistic rules not involved in inconsistency. A semantic characterization of these revision operators in terms of possibilistic distribution is then provided. Logical properties of the proposed operators are presented as well as a computational complexity study.

2 Background

2.1 Possibilistic Answer Set Programming

In this subsection, we recall the definitions about possibilistic answer set programming that can be found in [Nicolas *et al.*, 2006].

Language of possibilistic ASP We consider given a finite set of atoms $\mathcal{X}$ and a finite, totally ordered set of necessity values $\mathcal{N} \subseteq]0,1]$. Let us first recall, that, as in possibilistic logic, these values are evaluated by a necessity measure and are not probabilities. Thus, they are not an absolute evaluation (like it is done in probability theory) but induce a certainty (or confidence) scale allowing to order beliefs. When the level of the belief is 1, it is considered as certain. Let us note that a piece of information with a necessity degree equal to 0 is not taken into account because it means that this piece is impossible.

A *possibilistic atom* is a pair $p = (x, \alpha) \in \mathcal{X} \times \mathcal{N}$. The set of atom sets is denoted by $2^{\mathcal{X}}$ and the set of possibilistic atom sets, in which every atom occurs at most once, is denoted by $\mathcal{A}$. We denote by $p^* = x$ the classical projection of p and by $n(p) = \alpha$ its necessity degree. Projection is extended to possibilistic atom sets in the usual way. A possibilistic ASP program is a finite set of rules r such that:

$$r = (c \leftarrow a_1,\ \ldots,\ a_n,\ \mathit{not}\, b_1,\ \ldots,\ \mathit{not}\, b_m,\ \alpha)$$

where $n \geq 0, m \geq 0$, $c,\ a_1,\ \ldots,\ a_n,\ b_1,\ \ldots,\ b_m \in \mathcal{X}$ and $\alpha \in \mathcal{N}$. We denote by $r^* = (c \leftarrow a_1,\ \ldots,\ a_n,\ \mathit{not}\, b_1,\ \ldots,\ \mathit{not}\, b_m)$ the classical projection of the rule r and $n(r) = \alpha$ its necessity degree representing the certainty level of the information described by the rule. The set of possibilistic ASP programs, in which every (classical) rule occurs at most once, is denoted by $\mathcal{P}$. The weight of $P \in \mathcal{P}$ is defined by $w(P) = max\{\alpha \mid (r, \alpha) \in P\}$.

A rule of the form $r = (c \leftarrow a_1,\ \ldots,\ a_n, \alpha)$ is called a possibilistic definite rule. When a program contains only possibilistic definite rules, it is called a possibilistic definite logic program. The set of possibilistic definite logic programs is denoted by $\mathcal{P}_d$.

We recall some operations over possibilistic ASP programs useful in the following. Let $P, Q \in \mathcal{P}$
$P^* = \{r \mid (r, \alpha) \in P\}$.
$P \sqcup Q = \{(r, \alpha) \mid (r, \alpha) \in P, r \notin Q^*\} \cup \{(r, \beta) \mid r \notin P^*, (r, \beta) \in Q,\} \cup \{(r, max\{\alpha, \beta\}) \mid (r, \alpha) \in P, (r, \beta) \in Q\}$.
$P \setminus^p Q = \{(r, \alpha) \mid (r, \alpha) \in P, (r \notin Q^*)$ or $(\exists \beta \mid (r, \beta) \in Q$ and $\beta < \alpha)\}$
For a classical (without weight) rule r, we use the following notations: $body^+(r) = \{a_1,\ \ldots,\ a_n\}$ is the positive body, $body^-(r) = \{b_1,\ \ldots,\ b_m\}$ is the negative body, $head(r) = c$ is the head and $r^+ = head(r) \leftarrow body^+(r)$ is the positive projection of the rule. These functions, defined for rules, are extended to rule sets as usual.

Possibilistic answer sets As in the classical case without necessity value, the definition of the answer sets is given by using the reduct of a program.

Definition 1. *Let $P \in \mathcal{P}$ and $A \in 2^{\mathcal{X}}$. The* possibilistic reduct *of P w.r.t. A is the possibilistic definite logic program $P^A = \{(r^{*+}, n(r)) \mid r \in P, body^-(r^*) \cap A = \emptyset\}$.*

Definition 2. *Let $r = (c \leftarrow a_1,\ \ldots,\ a_n, \alpha)$ be a definite possibilistic rule ($body^-(r^*) = \emptyset$) and $A \in \mathcal{A}$, r is β-applicable in A if $\{(a_1, \alpha_1), \ldots, (a_n, \alpha_n)\} \subseteq A$ and $\beta = \min\{\alpha, \alpha_1, \ldots, \alpha_n\}$.*

For a given $P \in \mathcal{P}_d$ and $A \in 2^{\mathcal{X}}$, $App(P, A) = \{r \in P^* \mid body^+(r) \subseteq A\}$.

Definition 3. *Let $P \in \mathcal{P}_d$ and $A \in \mathcal{A}$. The* immediate possibilistic consequence operator *ΠT_P is a function from $\mathcal{A}$ to $\mathcal{A}$ such that:*
$\Pi T_P(A) = \{(x, N(x, A, P)) \mid x \in head(App(P, A^*))\}$ *with*
$$N(x, A, P) = \max_{r \in P}\{\nu \mid r \textit{ is } \nu\textit{-applicable in } A \textit{ and } x = head(r^*)\}$$
then the iterated operator ΠT_P^k is defined by
$$\Pi T_P^0 = \emptyset \quad \textit{and} \quad \Pi T_P^{n+1} = \Pi T_P(\Pi T_P^n), \forall n \geq 0.$$

Proposition 1 ([Nicolas *et al.*, 2006])**.** *Let $P \in \mathcal{P}_d$, then ΠT_P has a least fix-point $\sqcup_{n \geq 0} \Pi T_P^n$ that we called the set of* possibilistic consequences *of P and we denote it by $\Pi Cn(P)$.*

By this way, the definition of a possibilistic answer set is natural.

Definition 4. *Let $P \in \mathcal{P}$ and $S \in \mathcal{A}$. S is a possibilistic answer set of P if $S = \Pi Cn(P^{(S^*)})$.*

The set of all possibilistic answer sets of P is denoted by $\Pi AS(P)$. Note that there is a one to one correspondence between the possibilistic answer sets of a possibilistic program P and the answer sets of its projection P^*.

Possibility distribution From a semantic point of view, a possibility distribution is defined over interpretations w.r.t. a possibilistic logic program. The possibility degree of each interpretation is induced by the necessity degrees associated to the falsified rules. This possibility degree reflects the ability of the interpretation to be an answer set of the program.

Let $P \in \mathcal{P}$ and $A \in 2^{\mathcal{X}}$, then $Fal(P, A) = \{r \in P \mid body^+(r^*) \subseteq A,\ body^-(r^*) \cap A = \emptyset,\ head(r^*) \notin A\}$ denotes the set of the rules of a possibilistic program P that are falsified w.r.t. an atom set A.

A set of classical rules R is *grounded* if there exists an enumeration $\langle r_1 \dots r_n \rangle$ of the rules of R such that $\forall i \in [1..n]$, $body^+(r_i) \subseteq head\{r_j \mid j < i\}$.

Definition 5. *Let $P \in \mathcal{P}_d$, the least possibility distribution $\pi_P : 2^{\mathcal{X}} \rightarrow [0,1]$ is defined by: for each $A \in 2^{\mathcal{X}}$*

- *if $A \not\subseteq head(App(P,A))$ then $\pi_P(A) = 0$,*
- *if $App(P,A)$ is not grounded then $\pi_P(A) = 0$,*
- *if A is an answer set of P^* then $\pi_P(A) = 1$,*
- *otherwise $\pi_P(A) = 1 - w(Fal(P,A))$.*

The distribution defined on definite programs can be used to define a distribution on ASP programs:

Definition 6. *Let $P \in \mathcal{P}$, the possibility distribution $\tilde{\pi}_P : 2^{\mathcal{X}} \rightarrow [0,1]$ is defined by: for each $A \in 2^{\mathcal{X}}$*

$$\tilde{\pi}_P(A) = \pi_{P^A}(A).$$

The following proposition shows that the interpretations with a value of 1 match with the answer sets of the possibilistic program without weights.

Proposition 2 ([Nicolas *et al.*, 2006]). *Let $P \in \mathcal{P}$ and $A \in 2^{\mathcal{X}}$. $\tilde{\pi}_P(A) = 1$ if and only if $A \in \Pi AS(P)^*$.*

Consistency in possibilistic ASP We say that $P \in \mathcal{P}$ is *consistent* if $\Pi AS(P) \neq \emptyset$ or, equivalently, if $\exists A \in 2^{\mathcal{X}}$ such that $\tilde{\pi}_P(A) = 1$. Otherwise P is said *inconsistent*.

We can also consider monotonic consistency. Let $P \in \mathcal{P}$ and $A \in 2^{\mathcal{X}}$, A is a (classical) model of P if $Fal(P,A) = \emptyset$. The set of models of P is denoted by $Mod(P)$. $P \in \mathcal{P}$ is *m-consistent* if $Mod(P) \neq \emptyset$. Otherwise P is said *m-inconsistent*.

2.2 Notations

We review some 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. Equivalence is defined by $x \simeq y$ if $x \leq y$ and $y \leq x$. The corresponding strict total preorder, denoted by $<$, is the relation defined by $x < y$ 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: $Min(M, \leq) = \{x \in M \mid \nexists y \in M, y < x\}$.

Let X and Y be two sets, $|X|$ (resp. $|Y|$) denotes the cardinality of X (resp. of Y) and $X \leq_{Card} Y$ if $|X| \leq |Y|$.

Let A be a finite set, a selection function denoted by f is a function from $2^A \setminus \emptyset$ to A which for any non empty set $X \in 2^A$ returns an element $f(X)$ such that $f(X) \in X$.

3 Possibilistic ASP Base Revision

The belief revision principles state that new information has to be accepted, the revised agent's beliefs have to be consistent and the initial agent's beliefs have to be changed as little as possible. This section studies belief revision when agent's beliefs and new information are represented by possibilistic ASP programs. In the following, according to the revision principles, we assume that the rules of the program to be revised P are not certain, i.e. $w(P) < 1$. The input is a possibilistic ASP program Q which is assumed to be certain, i.e. $\forall r \in Q, n(r) = 1$. Moreover, we suppose that Q is m-consistent otherwise revising by Q is impossible.

We present two different rule-based revision strategies. The first one, focuses on keeping the most certain possibilistic rules of P beyond the consistency degree, while the second one focuses on removing a minimum number of the least certain rules of P, keeping the rules of P not involved in inconsistency.

3.1 Standard Possibilistic ASP Base Revision

The revision presented here follows the approach of [Nicolas *et al.*, 2006] to restore consistency of a possibilistic normal logic program by deleting less certain rules. In this view, if a rule (r, α) must be removed in order to restore consistency, then all the rules with a degree less or equal to α have to be removed too. Indeed, questioning a belief is questioning all beliefs with lower necessity.

We first recall some additional notions from [Nicolas *et al.*, 2006]. Let $P \in \mathcal{P}$ the α-cut of P, denoted by $P_{>\alpha}$, is such that $P_{>\alpha} = \{(r_i, \alpha_i) \in P \mid \alpha_i > \alpha\}$. The inconsistency degree of P, denoted by $Inc(P)$, is such that $Inc(P) = \min\{\alpha \mid \alpha \in \mathcal{N} \cup \{0\}, P_{>\alpha} \text{ is consistent}\}$. The inconsistency degree of P defines the minimum level of certainty for which a cut of P is consistent. But, it is not necessarily the greatest (in number of rules) consistent subprogram of P.

The *standard possibilistic ASP revision* operator, denoted by $\circledast_\Pi$, is a function from $\mathcal{P} \times \mathcal{P}$ to $\mathcal{P}$ defined as follows.

Definition 7. *Let $P, Q \in \mathcal{P}$ with $w(P) < 1$, $\forall r \in Q$, $n(r) = 1$ and Q is m-consistent.*

$$P \circledast_\Pi Q = (P \sqcup Q)_{>Inc(P \sqcup Q)}$$

The following example illustrates the behavior of the $\circledast_\Pi$ revision operator.

Example 1. *Let $P = \{r_1 : (b \leftarrow not\, c,\ \ 0.9), r_2 : (d \leftarrow\ ,\ \ 0.8), r_3 : (c \leftarrow not\, a,\ \ 0.8), r_4 : (e \leftarrow,\ \ 0.7)\}$ and $Q = \{r_0 : (a \leftarrow not\, b,\ 1)\}$. The inconsistency of $P \sqcup Q$ comes from the rules $\{r_0, r_1, r_3\}$. We have $\Pi AS(P \sqcup Q) = \emptyset$ and $Inc(P \sqcup Q) = 0.8$, the revised possibilistic logic program is $P \circledast_\Pi Q = \{r_0, r_1\}$. Note that the rules r_2 and r_4 which do not contribute to the inconsistency of $P \sqcup Q$ do not appear in the revised logic program because they are less certain than the "guilty" rule r_3.*

Note that if $P \sqcup Q$ is consistent then $Inc(P \sqcup Q) = 0$ therefore $P \circledast_\Pi Q = P \sqcup Q$. If $Inc(P \sqcup Q) = 1$ then $P \circledast_\Pi Q = \emptyset$ and $\Pi AS(P \circledast_\Pi Q) = \{\emptyset\}$. That means that Q is inconsistent and that nothing can be done (with the present strategy) to restore the consistency as illustrated in the following example.

Example 2. *Let $P = \{(b \leftarrow, 0.8)\}$ and $Q = \{(\leftarrow not\, a, 1)\}$, we have $Inc(P \sqcup Q) = 1$ thus $P \circledast_\Pi Q = \emptyset$.*

When dealing with possibility theory where necessity degrees represent levels of certainty, it is natural to remove all the rules that are less certain that the rules responsible for inconsistency. However, if the weights represent priorities and if the goal is to remove the smallest number of rules w.r.t. priorities, it is necessary to define a less drastic revision strategy.

3.2 RSR Possibilistic ASP Base Revision

We extend Removed Sets Revision (RSR) [Hué *et al.*, 2013] to possibilistic ASP. This strategy focuses on the minimal number of possibilistic rules of least degree of necessity to remove in order to restore consistency. Here, only the rules involved in inconsistency are removed. We first introduce the notion of *possibilistic potential removed set*.

Definition 8 (possibilistic potential removed set)**.** *Let $P, Q \in \mathcal{P}$, a possibilistic potential removed set X is such that: (i) $X \subseteq P$. (ii) $(P \setminus X) \sqcup Q$ is consistent. (iii) For each $X' \subset X$, $(P \setminus X') \sqcup Q$ is inconsistent.*

$\mathcal{PR}(P,Q)$ denotes the set of possibilistic potential removed sets for P and Q. According to the definition, if $P \sqcup Q$ is consistent then $\mathcal{PR}(P,Q) = \{\emptyset\}$. Since only rules from P can be removed in order to restore consistency of $P \sqcup Q$, it may be possible that the set of potential removed sets for an inconsistent program Q is empty which is the case considering P and Q from Example 2.

In order to keep the rules with the highest degrees of necessity we define a total pre-order on possibilistic potential removed sets, denoted by $\leq_w$, based on the weight of a set of possibilistic rules: for all X, Y possibilistic potential removed sets, $X \leq_w Y$ if and only if $w(X) \leq w(Y)$. Moreover, we use the cardinality minimality criterion on possibilistic potential removed sets of minimal weight.

Definition 9 (possibilistic removed set)**.** *Let $P, Q \in \mathcal{P}$, a possibilistic removed set X is such that: (i) $X \in \mathcal{PR}(P,Q)$. (ii) $\nexists Y \in \mathcal{PR}(P,Q)$ such that $Y <_w X$. (iii) $\nexists Y \in \mathcal{PR}(P,Q)$ such that $Y \simeq_w X$ and $Y <_{Card} X$.*

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

Example 3. *(Example 1 continued) $\mathcal{PR}(P,Q) = \{X_1 = \{r_1 : (b \leftarrow not\, c,\ \ 0.9)\}, X_2 = \{r_3 : (c \leftarrow not\, a,\ \ 0.8)\}\}$, $w(X_2) < w(X_1)$ therefore $\mathcal{R}(P,Q) = \{X_2\}$.*

Let f be a selection function which chooses one removed set among the possible ones, the *Possibilistic Removed Set Revision* operator denoted by $\circledast_{RSR(f)}$ is a function from $\mathcal{P} \times \mathcal{P}$ to $\mathcal{P}$ defined as follows.

Definition 10. *Let $P, Q \in \mathcal{P}$ with $w(P) < 1$, $\forall r \in Q$, $n(r) = 1$ and Q is m-consistent.*

$$P \circledast_{RSR(f)} Q = (P \setminus f(\mathcal{R}(P,Q)))\ \sqcup\ Q.$$

Note that if $\mathcal{R}(P,Q) = \emptyset$ then $\circledast_{RSR(f)}$ is not defined. If $P \sqcup Q$ is consistent then $\mathcal{R}(P,Q) = \{\emptyset\}$ thus $P \circledast_{RSR(f)} Q = P \sqcup Q$.

Example 4. *(Example 1 continued) $\mathcal{R}(P,Q) = \{\{r_3 : (c \leftarrow not\, a,\ \ 0.8)\}\}$ and $P \circledast_{RSR(f)} Q = \{r_0 : (a \leftarrow not\, b,\ \ 1),\ r_1 : (b \leftarrow not\, c,\ \ 0.9),\ r_2 : (d \leftarrow,\ \ 0.8),\ r_4 : (e \leftarrow,\ \ 0.7)\}$. Note that there is no unnecessary removal.*

The following example illustrates several results of revision according to the chosen selection function.

Example 5. *Let P and Q such that $P = \{r_1 : (a \leftarrow not\ b,\ 0.9), r_2 : (b \leftarrow not\ c,\ 0.9)\}$ and $Q = \{r_0 : (c \leftarrow not\ a,\ 1)\}$. We have $\Pi AS(P) = \{\{b\}\}$, $\Pi AS(Q) = \{\{c\}\}$, $\Pi AS(P \sqcup Q) = \emptyset$ and $P \circledast_\Pi Q = Q$. However $\mathcal{R}(P,Q) = \{\{r_1\},\{r_2\}\}$. Let $f_1(\mathcal{R}(P,Q)) = \{r_1\}$ and $f_2(\mathcal{R}(P,Q)) = \{r_2\}$, we have $P \circledast_{RSR(f_1)} Q = \{r_0, r_2\}$ and $P \circledast_{RSR(f_2)} Q = \{r_0, r_1\}$.*

4 Semantic Characterization

4.1 Standard Possibilistic ASP Base Revision

We first recall the semantic approach of possibilistic ASP [Nicolas *et al.*, 2006] to deal with inconsistent programs. An inconsistency degree is defined in a semantic way, by using a possibilistic distribution.

Definition 11. *Let $P \in \mathcal{P}$, the inconsistency degree of P is*

$$SemInc(P) = 1 - \max_{A \in 2^{\mathcal{X}}} \{\tilde{\pi}_P(A)\}.$$

This inconsistency degree is used to define a cut that computes the greatest (w.r.t. the certainty level of rules) consistent subprogram of P.

Definition 12. *Let cut be the function defined from $\mathcal{P}$ to $\mathcal{P}$ by:*

- *$cut(P) = P$ if $SemInc(P) = 0$*
- *$cut(P) = cut(P_{>SemInc(P)})$ otherwise.*

The following proposition gives the semantic counterpart of the ASP base revision operator $\circledast_\Pi$.

Proposition 3. *Let $P, Q \in \mathcal{P}$ such that $w(P) < 1$, $\forall r \in Q$, $n(r) = 1$ and Q is m-consistent.*

$$P \circledast_\Pi Q = cut(P \sqcup Q).$$

This proposition directly follows from Definition 7 ($P \circledast_\Pi Q = (P \sqcup Q)_{>Inc(P \sqcup Q)}$) and from the result of [Nicolas *et al.*, 2006]: let $P \in \mathcal{P}$, $cut(P) = P_{>Inc(P)}$.

The following example illustrates the computation of the revised program by the *cut* function.

Example 6. *(Example 1 continued) We recall that $Q = \{r_0 : (a \leftarrow not\, b,\ \ 1)\}$ et $P = \{r_1 : (b \leftarrow not\, c,\ \ 0.9), r_2 : (d \leftarrow\ ,\ \ 0.8), r_3 : (c \leftarrow not\, a,\ \ 0.8), r_4 : (e \leftarrow,\ \ 0.7)\}$.*

In the least specific possibility distribution, the greatest possibility degree is 0.2 and corresponds to four interpretations:

- *$A_1 = \{b\}$. $Fal(P \sqcup Q, A_1) = \{r_2, r_3, r_4\}$, $w(Fal(P \sqcup Q, A_1)) = 0.8$ and $\tilde{\pi}(\{b\}) = 1 - 0.8 = 0.2$.*
- *$A_2 = \{b, d\}$. $Fal(P \sqcup Q, A_2) = \{r_3, r_4\}$, $w(Fal(P \sqcup Q, A_2)) = 0.8$ and $\tilde{\pi}(\{b, d\}) = 0.2$.*
- *$A_3 = \{b, e\}$. $Fal(P \sqcup Q, A_3) = \{r_2, r_3\}$, $w(Fal(P \sqcup Q, A_3)) = 0.8$ and $\tilde{\pi}(\{b, e\}) = 0.2$.*
- *$A_4 = \{b, d, e\}$. $Fal(P \sqcup Q, A_4) = \{r_3\}$, $w(Fal(P \sqcup Q, A_4)) = 0.8$ and $\tilde{\pi}(\{b, d, e\}) = 0.2$.*

The inconsistency degree is
$SemInc(P \sqcup Q) = 1 - \max_{A \in 2^{\mathcal{X}}} \{\tilde{\pi}_P(A)\} = 1 - 0.2 = 0.8$.
The first cut removes r_2, r_3 and r_4 and the resulting program $(P \sqcup Q)_{>0.8} = \{r_0, r_1\}$ is consistent. Note that it matches with the revised syntactic program $P \circledast_\Pi Q$.

The cut can be slightly harder to compute. Indeed, in the next example, two successive cuts are necessary to restore consistency.

Example 7. *Let $Q = \{r_1 : (a \leftarrow not\, d,\quad 1), r_2 : (a \leftarrow not\, b,\quad 1)\}$ et $P = \{r_3 : (b \leftarrow not\, c,\quad 0.9), r_4 : (c \leftarrow not\, a,\quad 0.8), r_5 : (d \leftarrow not\, e,\quad 0.7), r_6 : (e \leftarrow not\, a,\quad 0.6), r_7 : (b \leftarrow,\quad 0.5)\}$. The inconsistency of $P \sqcup Q$ comes from the rules $\{r_1, r_5, r_6\}$. The inconsistency degree is 0.6. It corresponds to the interpretation $A = \{b, c, d\}$ and $Fal(P \sqcup Q, A) = \{r_6\}$. But the first cut $(P \sqcup Q)_{>0.6}$ is inconsistent: the rule r_6 is removed but also the rule r_7 because its weight is lower (0.5). The removal of r_7 creates a new inconsistency with the cycle $\{r_2, r_3, r_4\}$ which was initially "blocked" by r_7. The inconsistency degree of $(P \sqcup Q)_{>0.6}$ is 0.7. It corresponds to the interpretation $A = \{a, b\}$ with $Fal((P \sqcup Q)_{>0.6}, A) = \{r_5\}$. A second cut removes r_5 and the resulting program $(P \sqcup Q)_{>0.7} = \{r_1, r_2, r_3, r_4\}$ is consistent.*

4.2 Possibilistic Removed Sets

We now present the semantic counterpart of the removed sets. It is based on the notions of preferred models and preferred falsified rules. In a possibility distribution for $P \sqcup Q$, the preferred models are those with the greatest possibility degree. If all interpretations have a degree equal to zero then the set of preferred models is empty.

Definition 13 (preferred models)**.** *Let $P, Q \in \mathcal{P}$, the set of preferred models $PrefMod(P, Q) = \{m \in Mod(Q) \mid \tilde{\pi}_{P \sqcup Q}(m) \neq 0$ and $\forall m' \in Mod(Q),\ \tilde{\pi}_{P \sqcup Q}(m') \leq \tilde{\pi}_{P \sqcup Q}(m)\}$.*

Example 8. *(Example 1 continued) In this example, A_1 to A_4 are the models of Q with the greatest possibility degree, thus $PrefMod(P, Q) = \{A_1, A_2, A_3, A_4\}$.*

Note that interpretations that are not models of Q have a degree of possibility equal to 0.

These models with the greatest possibility degree match with those for which the weight of the falsified rules is minimal. These sets of falsified rules are the "semantically preferred" ones (denoted by $PrefFal(P, Q)$) and the smallest (w.r.t inclusion and cardinality) of them correspond to the removed sets. The following proposition establishes this correspondence between syntactic and semantic definitions.

Proposition 4. *Let $P, Q \in \mathcal{P}$ such that $w(P) < 1$, $\forall r \in Q$, $n(r) = 1$ and Q is m-consistent. Let $PrefFal(P, Q) = \{Fal(P \sqcup Q, m) \mid m \in PrefMod(P, Q)\}$ be the sets of rules falsified by the preferred models,*

$$Min(Min(PrefFal(P, Q), \subseteq), \leq_{Card}) = \mathcal{R}(P, Q).$$

Sketch of proof. Let $AllFal(P, Q) = \{X \mid X = Fal(P \sqcup Q, m), m \in Mod(Q), \tilde{\pi}_{P \sqcup Q}(m) \neq 0\}$. A first part of the proof establishes that $\Pi AS((P \setminus Fal(P, m)) \sqcup Q)^* = \{m \in Mod(Q) \mid m \subseteq head(App((P \sqcup Q)^m, m))$ and $App((P \sqcup Q)^m, m)$ is grounded$\}$. This result allows one to link Definition 8 (of a potential removed set X based on the condition that $(P \setminus X) \sqcup Q$ is consistent), and the semantic characterization based on the falsified rules in the models such that $\tilde{\pi}_{P \sqcup Q}(m) \neq 0$ (see conditions in Definitions 5 and 6 of the possibility distribution). It is then established that $\mathcal{PR}(P, Q) = Min(AllFal(P, Q), \subseteq)$ and finally that $Min(\mathcal{PR}(P, Q), \leq_w) = Min(PrefFal(P, Q), \subseteq)$. □

The following examples illustrate the semantic characterization of the removed sets.

Example 9. *(Example 1 continued) $PrefFal(P, Q) = \{\{r_2, r_3, r_4\}, \{r_3, r_4\}, \{r_2, r_3\}, \{r_3\}\}$ and $Min(Min(PrefFal(P, Q), \subseteq), \leq_{Card}) = \{\{r_3\}\} = \mathcal{R}(P, Q)$.*

Example 10. *(Example 7 continued) Here, $PrefMod(P, Q) = \{\{b, c, d\}\}$, $PrefFal(P, Q) = \{\{r_6\}\}$ and $Min(Min(PrefFal(P, Q), \subseteq), \leq_{Card}) = \{\{r_6\}\} = \mathcal{R}(P, Q)$. Note that the result is direct contrary to the standard possibilistic revision where two cuts are necessary.*

The semantic characterization of $\circledast_{RSR(f)}$ immediately follows from the semantic characterization of the removed sets.

Furthermore if all rules of P have the same weight $\alpha \in]0, 1[$ then all the models of Q with a non-zero degree ($\tilde{\pi}_{P \sqcup Q}(m) \neq 0$) have the same degree $1 - \alpha$ and the falsified rules corresponding to these models (the $AllFal(P, Q)$ set from the proof sketch) match with the *canonical removed sets* from [Garcia *et al.*, 2017]. Thus the present work is a generalization of the approach of [Garcia *et al.*, 2017].

5 Logical Properties

In this section we go a step further in the characterization of Possibilistic ASP base revision by presenting logical properties of the proposed operators through a set of postulates. Hansson's postulates [Hansson, 1999] have been formulated to characterize belief base revision in a classical (monotonic) logic setting. We now adapt them within the non-monotonic possibilistic ASP framework.

Let P, Q, X, $R \in \mathcal{P}$, and $\star$ be a revision operator.

Success	$Q \subseteq P \star Q$.
Inclusion	$P \star Q \subseteq P \sqcup Q$.
Consistency	$P \star Q$ is consistent.
Vacuity	If $P \sqcup Q$ is consistent then $P \star Q = P \sqcup Q$.
Relevance	If $R \neq \emptyset$, $R \subseteq P \setminus^p (P \star Q)$ then $(P \star Q) \sqcup R$ is inconsistent.
Uniformity	If for all subsets X of P, $X \sqcup Q$ is consistent if and only if $X \sqcup R$ is consistent then $P \setminus^p (P \star Q) = P \setminus^p (P \star R)$.

The meaning of the postulates is the following: *Success* gives priority to new information. *Inclusion* states that the union of the initial logic programs is the upper bound of any revision operation. *Consistency* expresses that the result of revision is consistent. *Vacuity* establishes that if the input is consistent with the initial logic program then nothing is removed from it. *Relevance* expresses the intuition that nothing is removed from the original logic program unless its removal contributes in some way to make the result consistent. *Uniformity* determines that if two consistent logic programs are consistent with the same subsets of the initial logic program P then the respective erased set of rules of P should be identical.

Proposition 5. *$\circledast_{\Pi}$ satisfies* Inclusion*,* Consistency *and* Vacuity *but it does not satisfy* Success*,* Relevance *nor* Uniformity.

Proof. By definition $\circledast_\Pi$ satisfies *Inclusion*, *Consistency* and *Vacuity* . *Success*: a counterexample is given in Example 2. *Relevance*: a counterexample is given in Example 1. We have $P \circledast_\Pi Q = \{r_0, r_1\}$. Let $R = \{r_4\}$, $R \subseteq P \setminus^p (P \circledast_\Pi Q)$ but $(P \circledast_\Pi Q) \sqcup R = \{r_0, r_1, r_4\}$ is consistent. This is due to the removal of rules that are not involved in inconsistency but for which the certainty degree is low.
Uniformity: we give a counterexample: $P = \{(a \leftarrow \mathit{not}\ b,\ 0.9), (c \leftarrow \mathit{not}\ a,\ 0.7), (e,\ 0.5), (f,\ 0.5)\}$, $Q = \{(e,\ 1), (b \leftarrow \mathit{not}\, c,\ 1)\}$, $R = \{(f,\ 1), (b \leftarrow \mathit{not}\, c,\ 1)\}$. We have $P \circledast_\Pi Q = \{(e,\ 1), (b \leftarrow \mathit{not}\ c,\ 1), (a \leftarrow \mathit{not}\ b,\ 0.9)\}$ and $P \circledast_\Pi R = \{(f,\ 1), (b \leftarrow \mathit{not}\ c,\ 1), (a \leftarrow \mathit{not}\ b,\ 0.9)\}$ thus $P \setminus^p (P \circledast_\Pi Q) = \{(c \leftarrow \mathit{not}\ a,\ 0.7), (f,\ 0.5)\}$ and $P \setminus^p (P \circledast_\Pi R) = \{(c \leftarrow \mathit{not}\ a,\ 0.7), (e,\ 0.5)\}$ therefore $P \setminus^p (P \circledast_\Pi Q) \neq P \setminus^p (P \circledast_\Pi R)$. □

Proposition 6. *If $\circledast_{RSR(f)}$ is defined, for any selection function f, $\circledast_{RSR(f)}$ satisfies* Success*,* Inclusion*,* Consistency*,* Vacuity *,* Relevance*, and* Uniformity*.*

Sketch of proof. By definition if $\circledast_{RSR(f)}$ is defined, $\circledast_{RSR(f)}$ satisfies *Success*, *Inclusion*, *Consistency* and *Vacuity*.
Relevance: For any selection function f, we have $P \setminus^p (P \circledast_{RSR(f)} Q) = P \setminus^p ((P \setminus f(\mathcal{R}(P, Q))) \sqcup Q) = f(\mathcal{R}(P, Q))$ by definition of the operators $\sqcup$ and $\setminus^p$, by the hypothesis $w(P) < 1$ and $\forall r \in Q$, $n(r) = 1$ and since $f(\mathcal{R}(P, Q))$ is a removed set. Thus, for any selection function f, if $R \subseteq P \setminus^p (P \circledast_{RSR(f)} Q)$ then $R \subseteq f(\mathcal{R}(P, Q))$ and, by Definition 9, $(P \circledast_{RSR(f)} Q) \sqcup R$ is inconsistent.
Uniformity : We can first easily prove that if for all subsets X of P, $X \sqcup Q$ is consistent if and only if $X \sqcup R$ is consistent then $\mathcal{PR}(P, Q) = \mathcal{PR}(P, R)$. For any selection function f, we have $P \setminus^p (P \circledast_{RSR(f)} Q) = f(\mathcal{R}(P, Q))$ and $P \setminus^p (P \circledast_{RSR(f)} R) = f(\mathcal{R}(P, R))$. Since $\mathcal{PR}(P, Q) = \mathcal{PR}(P, R)$ the result holds. □

6 Complexity

We study the following complexity problem:
Name : ΠASPModelChecking(RS)
Input : $P, Q \in \mathcal{P}$, $X \in \mathcal{A}$.
Question : $\exists R \in \mathcal{R}(P, Q)$ such that $X \in \Pi AS((P \setminus R) \sqcup Q)$?

Proposition 7. ΠASPModelChecking(RS) *is in* $\mathbf{DP}$.

Proof. We recall that a language L is in the class $\mathbf{DP}$ if and only if there are two languages $L_1 \in \mathbf{NP}$ and $L_2 \in \mathbf{coNP}$ such that $L = L_1 \cap L_2$ [Papadimitriou, 1994][1].

Finding such a set R can be broken down as follows: 1. Is there a set $R \subseteq P$ such that $X \in \Pi AS((P \setminus R) \sqcup Q)$: (a) Guess a set of possibilistic normal rules R; (b) Check that $R \subseteq P$; (c) Check that $X \in \Pi AS((P \setminus R) \sqcup Q)$. 2. Compute $k_R = |R|$. 3. Check that $R \in \mathcal{R}(P, Q)$: (a) Guess a set of possibilistic normal rules R_0 and a set of possibilistic atoms X_0; (b) Check that $R_0 \subseteq P$; (c) Check that $R_0 <_w R$ or ($R_0 =_w R$ and $|R_0| < k_R$); (d) Check that $X_0 \in \Pi AS((P \setminus R_0) \sqcup Q)$.

The algorithmic difficulty concentrates in points 1 and 3. The algorithm described in point 1 can be solved in polynomial time on a non-deterministic Turing machine, and R is a certificate, because it succinctly proves that $X \in \Pi AS((P \setminus R) \sqcup Q)$. Thus, this subproblem is in $\mathbf{NP}$. In the algorithm described in point 3, (R_0, X_0) is a succinct disqualification, that is, it proves that $X \notin \mathcal{R}(P, Q)$. This algorithm can run in polynomial time on a non deterministic Turing machine. Thus, it is in $\mathbf{coNP}$. □

Proposition 8. ΠASPModelChecking(RS) *is* $\mathbf{DP}$*-complete.*

Sketch of proof. The proof is based on a transformation of the problem ExactIndependentSet into two possibilistic normal logic programs P and Q, with a one to one correspondence between the possibilistic answer sets of this program and the maximal independent sets of size k of a graph $G = (V, E)$. We recall the definition of the problem ExactIndependentSet:
Name : ExactIndependentSet
Input : a graph $G = (V, E)$, a positive integer $k \leq |V|$.
Question : Does G contain an independent set of size k, i.e. a subset $V' \subseteq V$ such that $|V'| = k$ and such that no two vertices in V' are joined by an edge in E, such that there is no other independent set with a size $k' > k$.

The program Q contains: (a) rules describing the graph vertices and edges, and (b) rules defining what an independent set is. The program P asserts that all the vertices of the graph form an independent set. In this context, removed sets represent sets of vertices which have to be dropped, so the remaining vertices form an independent set. □

7 Conclusion

The paper addresses possibilistic ASP base revision with certain input. It proposes two revision operators. The first one, $\circledast_\Pi$, is fully defined in the spirit of possibility theory while the second one $\circledast_{RSR(f)}$ stemming from "removed sets revision" keeps the rules not involved in inconsistency. For both revision operators a semantic characterization is provided in terms of a possibilistic distribution and logical properties are provided in terms of satisfaction of Hansson's postulates rephrased within the possibilistic ASP framework. Moreover the computational complexity of the $\circledast_{RSR(f)}$ operator is given.

There are several issues to address as future work. The first one is to study the complexity of the $\circledast_\Pi$ operator.

In this paper, all rules of the revising logic program are certain, thus a natural future work is to extend the proposed operators to the case where the input is uncertain, leading to the definition of semi-revision operators. A strategy has to be defined when the necessity degree of some rules in the revising program is lower than the one of the same rules appearing in the initial program.

Another issue is to investigate other revision strategies stemming from possibilistic revision or ASP base revision.

Finally, possibilistic ASP base revision could be investigated within the framework of other possibilistic ASP extensions.

[1] The $\mathbf{DP}$ class is sometimes also called BH_2.

References

[Alchourrón and Makinson, 1985] C. E. Alchourrón and D. Makinson. On the logic of theory change : Safe contraction. *Studia Logica*, 44(4):14–37, 1985.

[Baget *et al.*, 2016] J.-F. Baget, Z. Bouraoui, F. Nouioua, O. Papini, S. Rocher, and E. Würbel. ∃-ASP for computing repairs with existential ontologies. In *Proc. of SUM'16*, pages 230–245, 2016.

[Baget *et al.*, 2018] J.-F. Baget, L. Garcia, F. Garreau, C. Lefèvre, S. Rocher, and I. Stéphan. Bringing existential variables in answer set programming and bringing nonmonotony in existential rules: two sides of the same coin. *Annals of Mathematics and Artificial Intelligence*, 82(1–3):3–42, 2018.

[Baral *et al.*, 2009] C. Baral, M. Gelfond, and J. N. Rushton. Probabilistic reasoning with answer sets. *Theory and Practice of Logic Programming*, 9(1):57–144, 2009.

[Bauters *et al.*, 2010] K. Bauters, S. Schockaert, M. De Cock, and D. Vermeir. Possibilistic answer set programming revisited. In *Proc. of UAI'10*, pages 48–55, 2010.

[Bauters *et al.*, 2015] K. Bauters, S. Schockaert, M. De Cock, and D. Vermeir. Characterizing and extending answer set semantics using possibility theory. *Theory and Practice of Logic Programming*, 15(1):79–116, 2015.

[Benferhat and Bouraoui, 2017] S. Benferhat and Z. Bouraoui. Min-based possibilistic DL-Lite. *Journal of Logic and Computation*, 27(1):261–297, 2017.

[de Morais and Finger, 2013] E. Menezes de Morais and M. Finger. Probabilistic answer set programming. In *Proc. of Brazilian Conference on Intelligent Systems, BRACIS'13*, pages 150–156, 2013.

[Delgrande *et al.*, 2013] J. P. Delgrande, P. Peppas, and S. Woltran. AGM-style belief revision of logic programs under answer set semantics. In *Proc. of LPNMR'13*, pages 264–276, 2013.

[Dubois and Prade, 1988] D. Dubois and H. Prade. *Possibility theory*. Plenum Press, New-York, 1988.

[Dubois and Prade, 1998] D. Dubois and H. Prade. Possibility theory: Qualitative and quantitative aspects. *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, 1:169–226, 1998.

[Dubois and Prade, 2001] D. Dubois and H. Prade. Possibility theory, probability theory and multiple-valued logics: A clarification. *Annals of Mathematics and Artificial Intelligence*, 32(1-4):35–66, 2001.

[Eiter *et al.*, 2008] T. Eiter, G. Ianni, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining answer set programming with description logics for the semantic web. *Artificial Intelligence*, 172(12-13):1495–1539, 2008.

[Garcia *et al.*, 2017] L. Garcia, C. Lefèvre, O. Papini, I. Stéphan, and E. Würbel. A semantic characterization for ASP base revision. In *Proc. of SUM'17*, pages 334–347, 2017.

[Gelfond and Lifschitz, 1988] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proc. of ICLP'88*, pages 1070–1080, 1988.

[Hansson, 1999] S. O. Hansson. *A textbook of belief dynamics. Theory change and database updating*. Kluwer, 1999.

[Hué *et al.*, 2013] J. Hué, O. Papini, and E. Würbel. Extending belief base change to logic programs with ASP. In *Trends in Belief Revision and Argumentation Dynamics*, Studies in Logic. 2013.

[Katsuno and Mendelzon, 1991] H. Katsuno and A. O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52(3):263–294, 1991.

[Krümpelmann and Kern-Isberner, 2012] P. Krümpelmann and G. Kern-Isberner. Belief base change operations for answer set programming. In *Proc. of JELIA'12*, pages 294–306, 2012.

[Lopes *et al.*, 2017] C. Lopes, M. Knorr, and J. Leite. Nohr: Integrating XSB prolog with the OWL 2 profiles and beyond. In *Proc. of LPNMR'17*, pages 236–249, 2017.

[Motik and Rosati, 2010] B. Motik and R. Rosati. Reconciling description logics and rules. *Journal of the ACM*, 57(5):30:1–30:62, 2010.

[Nicolas *et al.*, 2005] P. Nicolas, L. Garcia, and I. Stéphan. Possibilistic stable models. In *Proc. of IJCAI'05*, pages 248–253, 2005.

[Nicolas *et al.*, 2006] P. Nicolas, L. Garcia, I. Stéphan, and C. Lefèvre. Possibilistic uncertainty handling for answer set programming. *Annals of Mathematics and Artificial Intelligence*, 47(1–2):139–181, 2006.

[Papadimitriou, 1994] C. H. Papadimitriou. *Computational Complexity*, chapter 17, page 412. Addison-Wesley, 1994.

[Schaub, 2008] T. Schaub. Here's the beef: Answer set programming ! In *Proc. of ICLP'08*, pages 93–98, 2008.

[Slota and Leite, 2014] M. Slota and J. Leite. The rise and fall of semantic rule updates based on SE-models. *Theory and Practice of Logic Programming*, 14(6):869–907, 2014.

[Turner, 2003] H. Turner. Strong equivalence made easy: nested expressions and weight constraints. *Theory and Practice of Logic Programming*, 3:609–622, 2003.

[Wang and Lee, 2015] Y. Wang and J. Lee. Handling uncertainty in answer set programming. In *Proc. of AAAI'15*, pages 4218–4219, 2015.

[Zhuang *et al.*, 2016] Z. Zhuang, J. P. Delgrande, A. C. Nayak, and A. Sattar. Reconsidering agm-style belief revision in the context of logic programs. In *Proc. of ECAI'16*, pages 671–679, 2016.