Non Prioritized Answer Set Revision †

Gerardo I. Simari ‡ Marcelo A. Falappa

Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA)*
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur. Av. Alem 1253, (8000) Bahía Blanca, Argentina
Tel: ++54 291 459 5135 - Fax: ++54 291 459 5136
{gis,mfalappa}@cs.uns.edu.ar

Abstract

In this paper, we build on previous work on Belief Revision operators based on the use of logic programming with Answer Set semantics as a representation language. We present a set of postulates for Answer Set Revision with respect to a set of sentences and with respect to *explanations*. We focus on the non-prioritized revision operator with respect to explanations, or arguments, which is intended to model situations in which agents revise their knowledge as a result of dialogues with other agents in a multi-agent setting.

Keywords: Belief Change, Knowledge Representation, Logic Programming, Answer Set Semantics, AnsProlog*.

1 Introduction and Motivation

In Multi-agent Systems in particular, we are concerned with the way in which autonomous agents maintain their beliefs. This is a very important aspect of agent design because behavior is greatly determined by the beliefs that are currently held. The areas of *Knowledge Representation and Reasoning* and *Belief Revision* deal with the issues of how beliefs are represented, how they are used in reasoning processes, and how a current body of beliefs reacts to the appearance of new information. There are many problems that must be solved in order to effectively manage a body of knowledge which, more often than not, may be incomplete or even inconsistent. Reacting to perceptions from the environment, which can come from a simple sensor or from a dialogue with another agent, is an essential feature of autonomy, and an agent must always be prepared to revise its beliefs [SF03].

There have been many developments of logical formalisms to deal with the weaknesses that *Classic Logic* displays when used to cope with these problems. These proposals often carry the names of *Nonmonotonic*, or *Defeasible Reasoning*. Some examples of such models are

---

†Workshop de agentes y sistemas inteligentes (WASI)

‡Partially supported by *Comisión de Investigaciones Científicas* (CIC), Gobierno de la Provincia de Bs. As. Both authors partially supported by the *Agencia Nacional de Promoción Científica y Tecnológica* (PICT 13096) and the Secretaría de Ciencia y Tecnología of Universidad Nacional del Sur (24/N016).

*Member of the IICyTI (Instituto de Investigación en Ciencia y Tecnología Informática)

Reiter's Default Logic, Moore's Autoepistemic Logic, McCarthy's Circumscription, McDermott and Doyle's Nonmonotonic Logics, and Belief Revision (also called Belief Change). This last formalism was mainly introduced by Gärdenfors and later extended by Alchourrón, Gärdenfors, and Makinson [AGM85, Gä88].

Belief Revision has as its main objective to model the dynamics of knowledge, that is, the way in which an agent's knowledge must be updated when it finds new information. It is in the field of *Cognitive Robotics* where belief revision finds its most appropriate application. Agents (be them physical or software) must have the necessary flexibility in order to change their beliefs because they are the main motivators of their actions. There are many real life examples of human beings performing belief revision: while watching the news we might find out that our basketball team became champion, or a phone call from a friend could inform us that he has changed his job, or that he is no longer single. In this example, the basic operations of belief revision known as *expansion, contraction*, and *revision* are respectively mapped into finding out that our team is champion, that our friend is no longer single, and that he has changed jobs, respectively. These same ideas are also applicable to environments that are closer to the computational realm, such as the use of communication protocols where new protocols can be added (expansion), withdrawn (contraction), or a subset of them be replaced by one or more (revision).

Revisions are the most commonly used change operators because they allow a sentence α to be included into a set K, generating a new set K', preserving consistency in the new set. The traditional revision models are *prioritized*, that is, they give priority to new information over the information that is already part of their knowledge. This property does not seem plausible in the real world, because in many cases it is not reasonable to give priority to information just because it is new.

In *non prioritized* models, it is possible for new information *not* to be totally accepted. Such new information can be rejected or accepted only after a debate process. In this sense, there exists a variety of different non prioritized belief revision models, among which are David Makinson's Screened Revision [Mak97, Han97b], Sven Ove Hansson's semirevision operators [Han97a], André Fuhrmann's merge operations [Fuh97] and Falappa *et al.*'s [FKIS02] recently formulated revisions by sets of sentences. In this last work, a new kind of non prioritized revision operator based on the use of explanations is presented. It is argued that an agent, before incorporating information which is inconsistent with its knowledge, should request an explanation that supports this information. An explanation is characterized as being composed of an *explanans* (set of sentences supporting a certain belief), and an *explanandum* (the final conclusion).

An example of this situation is the following: Suppose a person believes that (α) *all felines can climb* and (β) *Moebius is a feline.* Thus, he will believe (δ) *Moebius can climb.* Later on, another person states that *Moebius cannot climb.* If δ is dropped, then α or β will have to be dropped as well. However, it does not seem rational to incorporate external beliefs without pondering them. An explanation should be required in order to incorporate such information, especially in the case in which it contradicts the previously maintained set of beliefs.

In our case, the person should demand an explanation for $\neg\delta$. One possibility is that he is given an explanation such as: *Moebius is a cat, but he cannot climb because he is hurt*, and *cats that are hurt cannot climb*. Now, the sentences in the explanans can be used to evaluate the new piece of information before it is incorporated. In [SF03], we proposed the use of AnsProlog* as a language for the representation of beliefs, as well as for explanations.

The rest of this work is organized as follows: in Section 2, we discuss belief revision with respect to sets of sentences and explanations. Section 3 includes a description of how AnsProlog* can be used as a knowledge representation language for belief revision based on the answer set semantics. Section 4 is the main part of the work; Sections 4.1 and 4.2 describe a set of postulates for revision with respect sets of sentences and explanations, respectively. Section 4.3 discusses how a belief revision operator based on the answer set semantics can be constructed[†], and finally, Section 5 includes an example of how this operator works.

2 Belief Revision by Sets of Sentences

An agent's beliefs can be represented by means of *belief sets* (sets of sentences closed under logical consequence) or *belief bases* (arbitrary sets of sentences). It is clear that, in computational applications, one must opt for finite belief bases. New information, which is called *epistemic input*, is sometimes represented by a sentence of the language or an arbitrary set of sentences.

The revision operator proposed in [FKIS02] allows a non prioritized revision in the following way:

- The epistemic input is a single sentence with an explanation for it.
- The explanation (a set of sentences with some constraints) is added to the original set, maybe resulting in a temporarily inconsistent set.
- Then the consistency is restored by a contraction by falsum.

It should be noted that this operator incorporates the possibility of *partial acceptance*, *i.e.*, even though the proposed explanation for α may be rejected, parts of it may still be added to the knowledge base in the process.

An *explanation* can be defined as follows [FKIS02]. The set A is an explanation for the sentence α if, and only if, the following conditions are satisfied,

1. Deduction: $A \vdash \alpha$.
2. Consistency: A $\nvdash \perp$
3. Minimality: if $B \subset A$ then $B \nvdash \alpha$.
4. Informational Content: $Cn(A) \not\subseteq Cn(\alpha)$.

[†]We thank Michael Gelfond for insightful conversations on the postulates and the construction of the Answer Set Revision operator.

Deduction guarantees that the explanans (support for a given belief) implies the explanandum (the belief being explained). Consistency prevents having an inconsistent explanation (which would explain *any* belief). Minimality establishes that every belief in the explanation is needed to obtain the explanandum, and Informational Content avoids cases in which the explanandum implies every sentence in the explanans. In particular, it avoids a sentence being an explanation for itself. As we can see from this description, and explanation can be seen as an external *argument* supporting a given belief.

Assume we want to revise a given belief base Π with respect to a given explanation A for a belief α. The revision involves:

1. Construction of counter-explanations for A from Π. These counter-explanations are minimal subsets of Π which are inconsistent with A.

2. A is compared with respect to its counter-explanations.

3. If A is (in some sense) "better" than its counter-explanations, then A is incorporated into Π and its counter-explanations (or part of them) are eliminated. In any other case, Π will remain unaltered.

The last step of the revision involves a decision between the proposed explanation, and the counter-explanations that can be built from Π. This decision will depend upon the particular (typically partial) ordering that is imposed over the set of beliefs. See Section 4.3 for a discussion of this point.

This mechanism could be applied in some real life situations, or dialogues between agents. In this paper, we build on previous work done by [SF03], where a new operator of this type, based on explanations was proposed. In the next section, we review the proposal of knowledge representation by means of AnsProlog* programs, and Section 4 includes a set of postulates that can be used to characterize this type of revision, and a construction is proposed.

3 An AnsProlog* Representation

We now consider the language of logic programming with respect to the answer set semantics, which is usually referred to as AnsProlog*. This name is short for "**Pro**gramming in **Log**ic with **Ans**wer sets" [GL90]. An AnsProlog* program is a finite collection of rules of the form:

$$L_0 \texttt{ or } \ldots \texttt{ or } L_k \leftarrow L_{k+1}, \ldots, L_m, \texttt{not } L_{m+1}, \ldots, \texttt{not } L_n.$$

where the L_i's are literals (in the sense of classical logic). Intuitively, a rule of this form means that if $L_{k+1}, \ldots, L_m$ are true and if $L_{m+1}, \ldots, L_n$ can be assumed to be false, then the sentence $L_0 \texttt{ or } \ldots \texttt{ or } L_k$ is true (*i.e.*, at least one of its literals is true). The symbol '*' in AnsProlog* means that no restriction is applied to the structure of the program's rules; a variety of syntactic sub-classes can be defined when such rules are restricted, as shown in [Bar03]. The importance of defining such a set of sub-classes lies in the varying degree of complexity of the rules in each class. This complexity may have a profound impact on the computational cost of operations that may be performed on programs, such as Answer Set checking and verifying if a given belief is entailed by a given program [Got94]. The study of how the computational complexity of these and other problems is related to belief dynamics under this representation is left as future work.

An AnsProlog* program can be used to represent an agent's beliefs by means of its *answer set semantics*. Such semantics can be defined as follows [GL02]:

1. A program Π *cautiously entails* a literal L ($\Pi \models L$) if L belongs to *all* answer sets of Π.

2. A program Π *bravely entails* a literal L ($\Pi \models_b L$) if L belongs to *some* answer sets of Π.

For programs having only *one* answer set, there is no difference between these two relations; as will be noted later, the Answer Set Revision operator is not affected by this choice in belief entailment. In the same way, any *explanation* for a given belief can be represented by an AnsProlog program; a subset of the program's rules can be interpreted as the explanans, and the desired explanandum will be entailed by the program. Because this program constitutes an explanation, it must satisfy the rules described in Section 2. Examples of belief bases and explanations will be shown in Section 5. In the next section, we introduce a non prioritized belief revision operator based on this representation.

4 Non Prioritized Answer Set Revision

An agent's belief base can be represented by an AnsProlog* program, and its beliefs will be reflected in the program's *answer set semantics* [GL02]. Therefore, a belief will belong to a given belief base if and only if it is entailed by its associated logic program. The answer sets shown in the previous section are examples of knowledge bases generated from logic programs. Because *explanations* can be seen as special cases of belief bases (they are sets of beliefs specifically designed to entail a given belief), they can be represented by AnsProlog* programs in the same way. It must be noted that, even though the answer set semantics yields a unique set, the complete set of answer sets is available at any time because it can be computed directly from the program. This fact plays an important role in the construction of the Answer Set Revision operator, as we will see below.

In this section, we will propose a characterization of a belief revision operator based on the use of AnsProlog* as a knowledge representation language, and the answer set semantics as we have seen in the previous sections. We will start with a set of postulates for revising a knowledge base with respect to a *set of sentences*, and then a set of postulates for revising a knowledge base with respect to an *explanation* will be proposed.

In the following, we will use the notation $AS(\Pi)$ to denote the *set of answer sets* for a program Π. In general, we will assume that a program has n answer sets, for some $n \geq 0$, and we will name each of these answer sets S_i, with $0 \leq i \leq n$.

4.1 Revision with respect to a *set* of sentences Γ

We propose the following set of postulates for the revision of a knowledge base Π with respect to a set of sentences Γ, by means of an answer set revision operator "$*$". In the context of answer set revision, we are revising a logic program with respect to another logic program.

Answer Sets are *consistent* sets of literals that obey the rules of programs. Therefore, if a program Π is inconsistent then $AS(\Pi) = \emptyset$. In general, we will be interested in knowledge bases Π such that $AS(\Pi) \neq \emptyset$.

Inclusion: For every belief α, such that $\alpha \in \Pi * \Gamma$, it holds that $\alpha \in \Pi$ or $\alpha \in \Gamma$.

The sentences that are included in the revised belief base are contained either in the original knowledge base or in the input set. This is also a modified version of a postulate in [FKIS02].

For example, assume $\Pi = \{a, b\}$, $\Gamma = \{c\}$, and $\Pi * \Gamma = \{a, b, c\}$. As this postulate states, for every one of a, b, or c, it holds that they are either contained in Π or in Γ.

Vacuity: If $AS(\Pi \cup \Gamma) \neq \emptyset$, then $\Pi * \Gamma = \Pi \cup \Gamma$.

If the input set is consistent with the knowledge base, then the revised knowledge base consists of the union of the original knowledge base and the input set.

Vacuity 2: If $\Gamma \subseteq \Pi$, then it holds that $\Pi * \Gamma = \Pi$

If all of the sentences contained in the input set are also contained in the knowledge base, then the knowledge base is not altered in the revision. This is a modified version of the same postulate that appears in [FKIS02].

Strong Success: If $AS(\Pi \cup \Gamma) \neq \emptyset$, then it holds that $\forall \alpha \in S_i, \forall S_i \in AS(\Gamma)$, $\alpha \in S_r$, for every $S_r \in AS(\Pi * \Gamma)$.

If the input set is consistent with the knowledge base, then *every* belief entailed by the input set is contained in *all* the Answer Sets of the revised knowledge base.

Weak Success: If $AS(\Pi \cup \Gamma) \neq \emptyset$, then it holds that $\forall \alpha \in S_i, \forall S_i \in AS(\Gamma), \alpha \in S_r$, for some $S_r \in AS(\Pi * \Gamma)$.

If the input set is consistent with the knowledge base, then *every* belief entailed by the input set is contained in *some* Answer Set of the revised knowledge base.

Stability: If $\Gamma \subseteq \Pi$, then it holds that $\Gamma \subseteq \Pi * \Gamma$.

If the beliefs contained in the input set are included in the beliefs contained in the knowledge base, then they will be included in the beliefs contained in the revised knowledge base.

For example, assume $\Pi = \{a, b, c\}$, $\Gamma = \{c\}$, and $\Pi * \Gamma = \{a, b, c\}$. As this postulate states, since $c \in \Gamma$ and $c \in \Pi$, then it holds that $c \in \Pi * \Gamma$.

Consistency: If $AS(\Gamma) \neq \emptyset$, then $AS(\Pi * \Gamma) \neq \emptyset$

If the explanation is consistent, then the revised knowledge base will be consistent.

Consistency Preservation: If $AS(\Pi) \neq \emptyset$, then $AS(\Pi * \Gamma) \neq \emptyset$.

If the knowledge base is consistent, then the revised knowledge base will be consistent.

Strong Consistency: $AS(\Pi * \Gamma) \neq \emptyset$.

The revised knowledge base will be consistent.

Congruence: If it holds that $\alpha \in \Pi \cup \Gamma$ if, and only if, $\alpha \in \Pi \cup \Delta$, then it holds that $\Pi * \Gamma = \Pi * \Delta$

If the union of the knowledge base with two different input sets is the same, then the respective revised knowledge bases with respect to these input sets are equal.

Weak Monotony: If $\Gamma \subseteq \Delta$, and $AS(\Pi \cup \Delta) \neq \emptyset$ then it holds that $\forall \alpha$ such that, if $\alpha \in \Pi * \Gamma$ then $\alpha \in \Pi * \Delta$.

If two input sets Γ and Δ are such that the sentences that are contained in Γ are also contained in Δ, and Δ is consistent with the knowledge base, then the beliefs contained in the knowledge base revised with respect to Γ are also contained in the knowledge base revised with respect to Δ.

4.2 Revision with respect to *explanations*

The following set of postulates are proposed for the revision of a knowledge base Π with respect to an explanation Ψ, by means of an answer set revision operator "$*$". This type of revision will model the situation in which an agent has received epistemic input, and an explanation (argument) that supports this input.

Weak Success: If Ψ is an explanation for α and $\neg\alpha \notin S_i$ $\forall S_i \in AS(\Pi)$ then $\alpha \in S_i, \forall S_i \in AS(\Pi * \Psi)$.

If the explanandum is consistent with *every* answer set of the knowledge base, then it will belong to *every* answer set of the revised knowlege base.

Cautious Success: If Ψ is an explanation for α and, for every β such that $\Psi \models \beta$, there is no $S_i \in AS(\Pi)$ such that $\neg\beta \in S_i$, then $\beta \in S_j$, for every $S_j \in AS(\Pi * \Psi)$.

If the explanation is consistent with *every* answer set of the knowledge base, then the beliefs that it entails will be included in *every* answer set of the revised knowledge base.

Brave Success: If Ψ is an explanation for α and, for every β such that $\Psi \models \beta$, there exists a set $S_i \in AS(\Pi)$ such that $\neg\beta \notin S_i$, then $\beta \in S_j$, for every $S_j \in AS(\Pi * \Psi)$.

If the explanation is consistent with *some* answer set of the knowledge base, then the beliefs that it entails will be included in *every* answer set of the revised knowledge base.

Vacuity for Explanations: If for every β such that $\Psi \models \beta$, it holds that $\beta \in S_i, \forall S_i \in AS(\Pi)$, then $\Pi * \Psi = \Pi$.

If the beliefs entailed by the explanation are included in *every* Answer Set of the knowledge base, then the revised knowledge base is equal to the original knowledge base.

Explanans Inclusion: If Ψ is an explanation for α and $\Psi \subseteq \Pi * \Psi$, then $\Pi * \Psi \models \alpha$.

If the explanation is included in the revised knowledge base, then the revised knowledge base entails the explanandum.

Expansion: If Ψ is an explanation for α and $\Pi \models \alpha$, then $\Pi * \Psi \models \alpha$.

If the original knowledge base entails the explanandum, then so will the revised knowledge base.

4.3 Construction

In this section, we will propose a construction for the Answer Set Revision operator, using the postulates discussed in the last section as guidelines. This construction is a variation of the one first presented in [SF03]. Current work is being directed towards establishing a formal relationship between this construction and the set of postulates.

When an agent with belief base Π is faced with an explanation Ψ for a given belief α, the agent must establish the status of the given information. There are three possible scenarios:

(*i*). α is consistent with *every* answer set of Π; *i.e.*, for every answer set $S \in AS(\Pi), \neg\alpha \notin S$.

(*ii*). α is only consistent with *some* of the answer sets of Π; *i.e.*, there is at least one answer set $S \in AS(\Pi)$ such that $\neg\alpha \notin S$.

(*iii*). α is *not* consistent with *any* of the answer sets of Π; *i.e.*, for every answer set $S \in AS(\Pi), \neg\alpha \in S$.

We will therefore construct an Answer Set Revision operator based on these cases:

- **Case 1:** *The explanation is consistent with* every *answer set of* Π.

  The new knowledge can be safely incorporated by adding Ψ to Π:

$$\Pi * \Psi \stackrel{def}{=} \Pi \cup \Psi$$

- **Case 2:** *The explanation is consistent with* some *answer sets in* Π.

  The existence of an answer set of Π that is inconsistent with Ψ means that one or more counter-explanations can be built from Π. Therefore, the proposed explanation must be evaluated with respect to the counter-explanations that can be constructed from Π. Such counter-explanations could be subsets of Π that have $\neg\alpha$ as their explanandum.

If the new knowledge is to be incorporated, the new knowledge base will be $\Pi * \Psi \stackrel{def}{=} \Pi \cup \Psi$ as in Case 1. This addition to Π will automatically remove those answer sets that were in disagreement with Ψ, and will incorporate new ones. In case Ψ is rejected, Π is left unaltered.

- **Case 3:** *The explanation is inconsistent with every answer set of* Π.

  This situation is, in some sense, the "opposite" of Case 1; it is generally the most difficult because the explanation presented is in complete disagreement with the beliefs in Π. As in Case 2, the explanation for the new information must be evaluated with respect to the counter-explanations that can be constructed from Π:

  – If Ψ is accepted, there are two sub-cases:

    * $AS(\Pi \cup \Psi) \neq \emptyset$. In this case, the revision can be solved as in Case 1, and $\Pi * \Psi \stackrel{def}{=} \Pi \cup \Psi$.
    * $AS(\Pi \cup \Psi) = \emptyset$. Here, the inconsistency *must be removed* from $\Pi \cup \Psi$.

  – If the agent rejects the explanation, the belief base will remain unaltered.

In Cases 2 and 3, it is necessary to have a way of *comparing* explanations with one another in order to decide if the explanation is preferable to the counter-explanation, or *viceversa*. We are currently studying how comparison criteria, such as [Poo85, SL92], can be formulated in our framework in order to specify how this decision is made.

Current work is also being dedicated to the study of how *kernel contractions* [Han94] can be constructed using this framework. In this way, in Case 3 we can contract the knowledge base Π with respect to the $\neg\alpha$-kernels, for every α such that $\Psi \models \alpha$, in order to eliminate the inconsistency.

As was mentioned earlier, the choice in semantics (cautious or brave) is irrelevant for this operator. This situation arises because the beliefs that could only be bravely entailed (and not cautiously entailed) are not considered to be "stable" beliefs, in the sense that beliefs α and $\neg\alpha$ could both be entailed by such semantics. An extension where this semantics is considered is left for future work.

We can intuitively see that this construction yields and operator that satisfies the postulates of Inclusion, Vacuity, Vacuity 2, Strong Consistency, and Stability from Section 4.1, and Vacuity for Explanations, and Explanans Inclusion from Section 4.2. In general, Expansion will not be satisfied.

5 An Example of Answer Set Revision with Explanations

In this section, we will show a simple example of the concepts introduced above. The domain is intentionally simple to keep the example from running too long.

Suppose an agent wishes to go on vacation, and that it is initially willing to make an investment of up to $2,500, so it goes to its usual travel agent and asks about possible destinations. Before going in search of options, its knowledge base is the following program Π:

$\perp \leftarrow buy\_ticket(X), price(X, PX), invest(M), greater(PX, M).$
$\perp \leftarrow invest(I), my\_money(M), greater(I, M).$
$\neg buy\_ticket(X) \leftarrow feature(X, Y), dislike\_feature(Y).$
$buy\_ticket(X) \leftarrow strong\_inclination(X).$
$\perp \leftarrow buy\_ticket(X), buy\_ticket(Y), X \neq Y.$

The first rule is called a *restriction*, and it states that the agent cannot buy X if its price is greater than the amount intended for investment. The second restriction, similar to this one, states that the agent cannot invest more money than it owns. The next rule represents that the agent does not wish to buy a ticket to a given place if it has a feature that it dislikes. Furthermore, the agent believes that it will buy a ticket to a destination if it is strongly inclined to visit this destination (fourth rule). Lastly, the agent wishes to purchase only one ticket. The belief base is completed with the following facts:

$invest(3000).\ my\_money(5000).\ strong\_inclination(ny).$
$dislike\_feature(expensive).\ dislike\_feature(bad\_weather).$

The first fact states that the agent is willing to invest \$2,500, and the second one declares that the agent actually has \$5,000. The third fact states that the agent is currently strongly inclined to go to New York for its vacations. The last two facts mean that the agent dislikes the respective features in a vacation destination.

The agent then goes to its travel agency, and the travel agent offers a trip to New York for \$2,800, or the alternative of staying in Argentina and going to Mar del Plata for \$500. Now, the agent has revised its beliefs; the knowledge base has been extended to include what the agent has observed. These inclusions can be directly made because they are in agreement with the only answer set of Π:

$price(ny, 2800).\ price(mdp, 500).$

Furthermore, the agent has also gathered information on each of the products:

$feature(ny, expensive).\ feature(ny, bad\_weather).\ feature(mdp, beach\_parties).$

New York turns out to be expensive, and the agent now has an inconsistency stemming from this situation: it wants to buy a ticket for New York because of its inclination, and it doesn't want to buy it because of the price. This inconsistency can be solved either by removing $strong\_inclination(ny)$, or replacing $invest(2500)$ with $invest(2800)$. Lastly, the agent has realized that, being summer in Argentina, the weather in New York will be very bad. The agent has also realized that Mar del Plata holds many beach parties this time of year. The agent has now made up its mind that it wants to buy one of the two tickets, and has included $buy\_ticket(ny)$ *or* $buy\_ticket(mdp)$ in its knowledge base.

Now, the agent has to make a decision between the two options; the travel agent gave good reasons to buy either ticket, building *explanations* that point out the advantages of the item being offered. The following explanation supports going to New York:

$$\begin{aligned}\Psi_{ny} = \{ \quad & buy\_ticket(X) \leftarrow advantage(X,Y), important\_feature(Y).,\\ & advantage(ny, sightseeing).,\\ & important\_feature(sightseeing).\}\end{aligned}$$

which minimally entails $buy\_ticket(ny)$. An explanation that supports going to Mar del Plata might look like this:

$$\begin{aligned}\Psi_{mdp} = \{ \quad & buy\_ticket(X) \leftarrow advantage(X,Y), important\_feature(Y),\\ & advantage(mdp, beach\_parties).,\\ & important\_feature(beach\_parties).\}\end{aligned}$$

However, the agent is able to build a *counter-explanation* for the explanation that supports going to New York. The counter-explanation for Ψ_{ny} is the following explanation for $\neg buy\_ticket(ny)$:

$$\begin{aligned}\Psi_{\neg ny} = \{ \quad & \neg buy\_ticket(X) \leftarrow feature(X,Y), dislike\_feature(Y).\\ & feature(ny, expensive),\\ & dislike\_feature(expensive).\}\end{aligned}$$

In order to gather the information on all the products, the agent initially rejected both explanations, and therefore its knowledge base has not been modified with respect to what to buy. The agent is now faced with a revision of its knowledge. Before going out, it expected to spend $2,500, even though it actually has $5,000 to spend. There are two possible situations that can arise after the agent revises its beliefs: it can either decide to purchase a ticket to New York, or it can opt for Mar del Plata. In the first case, the agent accepts the travel agent's offer, and its new knowledge base will entail $buy\_ticket(ny)$, which will involve removing $negative\_feature(expensive)$ from Π, and adjusting the amount it wishes to invest. In the second case, the agent decides to buy a ticket to Mar del Plata, and include $buy\_ticket(mdp)$ in its knowledge base; now, $strong\_inclination(ny)$ must be dropped. If this isn't done, there would be an inconsistency because both $buy\_ticket(ny)$ and $\neg buy\_ticket(ny)$ could be entailed.

6 Conclusions and Future Work

In this paper, we have built on previous work on belief revision based on logic programming with answer set semantics by introducing a set of postulates that characterize revision with respect to a *set* of sentences, and with respect to explanations. We have focused on the use of explanations in order to build an operator that is useful for agents that interact with their peers in multi-agent settings.

We are currently focusing on characterizing kernel contractions in our framework, and studying mechanisms to resolve the issue of preference among explanations (and counter-explanations). The topic of Computational Complexity is also regarded as important future research, because the computational cost of the various operations that can be performed on AnsProlog* programs is directly related to the computational cost of the defined operator. Existing efficient implementations of AnsProlog make this a promising line of research.

References

[AGM85] Carlos Alchourrón, Peter Gärdenfors, and David Makinson. *On the Logic of Theory Change: Partial Meet Contraction and Revision Functions*. *The Journal of Symbolic Logic*, 50:510–530, 1985.

[Bar03] C. Baral. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press, 2003.

[FKIS02] Marcelo A. Falappa, Gabriele Kern-Isberner, and Guillermo R. Simari. *Belief Revision, Explanations and Defeasible Reasoning*. *Artificial Intelligence Journal*, 141:1–28, 2002.

[Fuh97] André Fuhrmann. *An Essay on Contraction*. Studies in Logic, Language and Information, CSLI Publications, Stanford, California, 1997.

[Gä88] Peter Gärdenfors. *Knowledge in Flux: Modelling the Dynamics of Epistemic States*. The MIT Press, Bradford Books, Cambridge, Massachusetts, 1988.

[GL90] Michael Gelfond and Vladimir Lifschitz. Logic Program with Classical Negation. In David H. D. Warren and Peter Szeredi, editors, *Proceedings of the 7th Int. Conf. on Logic Programming*, pages 579–597. MIT, June 1990.

[GL02] Michael Gelfond and Nicola Leone. Logic programming and knowledge representation—the A-prolog perspective. *Artificial Intelligence*, 138(1–2):3–38, 2002.

[Got94] Georg Gottlob. Complexity and expressive power of disjunctive logic programming. In Maurice Bruynooghe, editor, *Logic Programming - Proceedings of the 1994 International Symposium*, pages 23–42, Massachusetts Institute of Technology, 1994. The MIT Press.

[Han94] Sven Ove Hansson. *Kernel Contraction*. *The Journal of Symbolic Logic*, 59:845–859, 1994.

[Han97a] Sven Ove Hansson. *Semi-Revision*. *Journal of Applied Non-Classical Logic*, 7:151–175, 1997.

[Han97b] Sven Ove Hansson. *Theoria: Special Issue on Non-Prioritized Belief Revision*. Department of Philosophy, Uppsala University, 1997.

[Mak97] David Makinson. *Screened Revision*. In **Theoria** [Han97b].

[Poo85] David Poole. *On the Comparison of Theories: Preferring the Most Specific Explanation*. In *Proceedings of IJCAI-85, Los Angeles (CA)*, pages 144–147, 1985.

[SF03] Gerardo I. Simari and Marcelo A. Falappa. Belief dynamics and explanations in ansprolog*. In Marcelo Naiouf, editor, *Proceedings of the IX Argentine Congress on Computer Science (CACIC)*, pages 589–600, La Plata, Buenos Aires, Argentina, 2003. Univ. Nac. de La Plata.

[SL92] Guillermo R. Simari and Ron P. Loui. *A Mathematical Treatment of Defeasible Reasoning and its Implementation*. *Artificial Intelligence*, 53:125–157, 1992.