Belief Dynamics and Defeasible Argumentation in Rational Agents*

Marcelo A. Falappa and **Alejandro J. García** and **Guillermo R. Simari**
Artificial Intelligence Research and Development Laboratory
Department of Computer Science and Engineering
Universidad Nacional del Sur
Av. Alem 1253, (8000) Bahía Blanca, Argentina

Abstract

The BDI model provides what it is possibly one of the most promising architectures for the development of intelligent agents, and has become one of the most studied and well known in the literature. The basic BDI model needs to be complemented with two mechanisms: one for reasoning about intentions, and one for revising beliefs upon perception.

In this work, we introduce a *Revision Operator by a Set of Sentences* which is a *non-prioritized* belief revision operator for changing the agent's beliefs. These beliefs are contained in a Knowledge Base which is represented using the language of Defeasible Logic Programming. This formalism provides a framework for knowledge representation and reasoning about beliefs and intentions. The Knowledge Base is in fact a defeasible logic program and the belief revision operator will transform this program preserving as much information as possible taking into account for this transformation the reasons offered to justify the change in beliefs.

Introduction

Understanding practical reasoning has been a goal in Artificial Intelligence since its beginnings (see (McCarthy 1958; McCarthy & Hayes 1969) and the collection (McCarthy 1990)). Lately, much attention has been focused in the definition and application of the concept of *Agency* (see (Huhns & Singh 1997; Weiss 1999; Jennings & Wooldridge 1998; Wooldridge & Rao 1999; Wooldridge 2000; Wooldridge & Jennings 1994)).

An intelligent agent is a physical or virtual entity in which certain general characteristics are recognized. It should be capable of acting on its environment in a flexible, autonomous manner, including the ability to communicate with similar entities. Furthermore, its behavior should be controlled by a set of tendencies. In designing agents with these characteristics, we need to devise an architecture in which the components of the agent are described and the interactions among these components are defined.

Practical reasoning can be described as the process of deciding what action to perform in order to reach the goals, and it involves two important processes: (a) Deliberation, which is the decision of *what* goals want to be reached, and (b) Means-ends reasoning: decide *how* the goals will be reached.

The *Belief-Desire-Intention* (BDI) model has its roots in the philosophical tradition of understanding practical reasoning. In the BDI model an agent has a set of *Beliefs*, a set of *Desires* and a set of *Intentions* (Bratman 1987). Intentions play a crucial role in the practical reasoning process. Perhaps the most obvious property of intentions is that they tend to lead to action. Intentions drive means-ends reasoning, constrain future deliberation, persist, and influence beliefs upon which future practical reasoning is based.

In order to design agents that have these (and other) desirable properties, a given model must be followed. The model's characteristics will greatly depend on the environment that the agent occupies, and on the type of behavior that is expected from it. These models carry the name of *architectures*, and they define a set of components that interrelate in order to generate the agent's behavior.

The BDI model provides what is possibly one of the most promising architectures for the development of intelligent agents, and it has become one of the most studied and well known in the literature (Rao & Georgeff 1991; 1992; Georgeff *et al.* 1999). An agent's intentions are a subset of the alternatives that are available to reach its goals. One of the most important characteristics that intentions have is their motivating role, *i.e.*, they provoke actions. Once an intention is adopted, it will affect the practical reasoning that goes on in the future. Intentions have the property of persistence, that is, in order for them to be useful, the agent must not abandon them. Intentions must persist until they are accomplished, until it is evident that they cannot be accomplished, or the reasons for their adoption are no longer valid. An agent's intentions are related to its beliefs about the future.

The basic BDI model needs to be complemented with two mechanisms: one for reasoning about intentions, and one for revising beliefs upon perception. In this work, we propose the use of Defeasible Logic Programming for knowledge representation and reasoning about beliefs and intentions, and we introduce a *non-prioritized* belief revision function that changes the agent's beliefs.

*Partially supported by Secretaría General de Ciencia y Tecnología de la Universidad Nacional del Sur and by Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 Nro 13096).

Agent's Knowledge Representation and Reasoning

Defeasible Logic Programming (abbreviated DeLP) will provide a representation language and a reasoning mechanism. Consequently, the agent's beliefs will be represented as a defeasible logic program. Here, we will introduce DeLP in an intuitive manner. The reader is referred to (García & Simari 2004; 1999) for a complete presentation of DeLP.

In DeLP, a program $\mathcal{P}$ is a pair (Π, Δ) where Π is a set of *facts* and *strict rules* and Δ a set of *defeasible rules*. Facts are represented by literals (ground atoms or negated ground atoms that use strong negation "$\sim$"), strict rules are denoted "$L_0 \leftarrow L_1, \ldots, L_n$", and defeasible rules are denoted"$L_0 \prec L_1, \ldots, L_n$". In both types of rules, the head L_0 is a literal, and the body $L_1, \ldots, L_n$ is a finite non-empty conjunction of literals. Defeasible rules are used to represent tentative information that may be used if nothing can be posed against it, whereas strict rules and facts represent non-defeasible knowledge. Thus, a defeasible rule represents a weak connection between the body and the head, and should be read as "reasons to believe in $L_1, \ldots, L_n$ provide reasons to believe in L_0". These rules, by representing weak connections, equip the representation language with a natural device to characterize a link between information that could be invalidated when more information comes into play (Simari & Loui 1992; Nute 1994).

DeLP also brings the possibility of representing negated facts and rules with heads containing negated literals. In that manner, the derivation of contradictory conclusions is possible. Since the set Π represents the non-defeasible part of the agent's beliefs, the agent's belief revision function should maintain the consistency of Π, and therefore contradictory literals cannot be derived from Π. However, since defeasible rules represent tentative information, defeasible derivations for contradictory literals are allowed from $\Pi \cup \Delta$. When this happens, a *defeasible argumentation* formalism is used for deciding which literal prevails as warranted.

Let's consider as an example a *printing agent* that has to select an appropriate printer upon user requirements. Its knowledge base would contain rules such as the following:

$$\Pi = \left\{ \begin{array}{l} \sim use(inkjet) \leftarrow use(laser) \\ \sim use(laser) \leftarrow use(inkjet) \end{array} \right\}$$

$$\Delta = \left\{ \begin{array}{l} use(inkjet) \prec file\_for\_printing \\ use(laser) \prec file\_for\_printing, high\_quality \end{array} \right\}$$

The strict rules of Π state that if one type of printer is selected then do not use the other. The first defeasible rule states that "if there is a file for printing, then there is a good (defeasible) reason to use an inkjet", whereas the second states that "if there is a file for printing in high quality, then there are good reasons for using a laser printer". Therefore, once the user requirements are provided, the agent may reason about what printer to select. The user requirements are represented by Π_1=$\Pi \cup \{file\_for\_printing\}$, then from (Π_1, Δ), the first defeasible rule will provide support for using the inkjet printer. Suppose that later the user requirements change and Π_2=$\Pi \cup \{file\_for\_printing, high\_quality\}$. From (Π_2, Δ) and the use of the first defeasible rule, an argument for selecting the inkjet could be constructed. Nevertheless, the second defeasible rule provides support for using the laser printer, which in turn allows to (defeasibly) infer not to use the inkjet. Therefore, the agent will face contradictory conclusions. This situation needs to be resolved, and the printing agent will use the defeasible argumentation mechanism that we will describe next.

In DeLP a literal L is *warranted* if there exists a non-defeated *argument* $\mathcal{A}$ supporting L. A set of defeasible rules $\mathcal{A}$ is an argument for a literal L, denoted $\langle \mathcal{A}, L \rangle$, if $\Pi \cup \mathcal{A}$ is a consistent set that entails L. In order to establish whether $\langle \mathcal{A}, L \rangle$ is a non-defeated argument, *argument rebuttals* or *counter-arguments* that could be *defeaters* for $\langle \mathcal{A}, L \rangle$ are considered, *i.e.*, counter-arguments that by some criterion are preferred to $\langle \mathcal{A}, L \rangle$. Since counter-arguments are arguments, there may exist defeaters for them, and defeaters for these defeaters, and so on. Thus, a sequence of arguments called an *argumentation line* is constructed, where each argument defeats its predecessor in the line. Going back to our printing agent example introduced above, from the program (Π_2,Δ), $\mathcal{A}$=$\{use(inkjet) \prec file\_for\_printing\}$ is an argument for $use(inkjet)$, and the argument $\mathcal{B}$=$\{use(laser) \prec file\_for\_printing, high\_quality\}$ is a defeater[1] for $\mathcal{A}$. Since $\mathcal{B}$ has no defeater, this is an argumentation line of two arguments, and $use(inkjet)$ is not warranted. However, since $\mathcal{B}$ has no defeaters, $use(laser)$ becomes warranted. Thus, the agent has the possibility of deciding among contradictory conclusions.

Usually, each argument has more than one defeater and therefore more than one argumentation line could exist. From the set of argumentation lines, a tree with arguments as nodes, called *dialectical tree*, is built. The root of the dialectical tree contains $\langle \mathcal{A}, L \rangle$ and each path from the root to a leaf represents an argumentation line. The tree is used for deciding whether L is warranted or not. This decision is taken after a *dialectical analysis* of the tree is perform. Notice that the construction of argumentation lines involves the detection and avoidance of fallacious argumentation lines. These fallacies, when unchecked, could lead to circularity and/or otherwise infinite sequences of arguments (Simari, Chesñevar, & García 1994; Prakken & Vreeswijk 2000). Thus, the dialectical tree is finite and a process for marking the nodes can be performed (see (García & Simari 2004)).

A leaf node of the tree is an argument with no defeaters and is marked as *undefeated*. An inner node is marked as *defeated* if it has at least an undefeated descendant, or is marked as *undefeated* if all its children are marked as defeated. A literal L is warranted if there exists an argument $\mathcal{A}$ for L and the dialectical tree for $\langle \mathcal{A}, L \rangle$ has its root node marked as undefeated.

This dialectical process could end in different ways. When looking for a warrant for a literal L, four different answers may result: YES, if there is a warrant for L; NO, if

[1] In this case the specificity criterion was used for comparing arguments. However, other criteria can be used (see (García & Simari 2004) for details)

there is a warrant for $\sim L$; UNDECIDED, if there is no warrant for L and no warrant for $\sim L$; and UNKNOWN, if L is a literal that is not possible to consider given the program $\mathcal{P}$=(Π, Δ), $i.e.$, L is not part of the language of $\mathcal{P}$ (see (García & Simari 2004)).

As stated above, the basic BDI model needs to be complemented with a mechanism for reasoning about intentions. In (Bratman 1987), Bratman *et al.* suggest the use of a "tractable system of defeasible reasoning" for reasoning about intentions. Here, we propose the use of DeLP for performing such task. In this approach, the agent's intentions will be represented as literals, and defeasible rules may be added to Δ for reasoning about these intentions. Since intentions are involved in action selection, we also propose the use of defeasible rules for this task. In this paper we will only consider atomic actions; however, this formalism may be easily extended for selecting pre-compiled plans. Plan formation, or plan reconsideration, is out of the scope of this paper, and is the subject of future research. We will assume that the agent has a perception function that provides the agent with a set of new facts and strict rules coming from the proper environment and/or other agents.

For example, consider a printer agent with a set of possible intentions such as:

$$I = \{use\_laser, use\_inkjet, use\_color\_inkjet\},$$

and the following knowledge base (Π, Δ):

$$\Pi = \left\{ \begin{array}{l} \sim use(inkjet) \leftarrow use(laser) \\ \sim use(color\_inkjet) \leftarrow use(laser) \\ \sim use(inkjet) \leftarrow use(color\_inkjet) \\ \sim use(laser) \leftarrow use(color\_inkjet) \\ \sim use(color\_inkjet) \leftarrow use(inkjet) \\ \sim use(laser) \leftarrow use(inkjet) \end{array} \right\}$$

$$\Delta = \left\{ \begin{array}{l} use(inkjet) \prec file\_for\_printing \\ use(laser) \prec file\_for\_printing, high\_quality \\ use(color\_inkjet) \prec file\_for\_printing, color \\ \sim use(X) \prec out\_of\_paper(X) \\ s\_a(s\_t(inkjet\_queue)) \prec use(inkjet) \\ s\_a(s\_t(laser\_queue)) \prec use(laser) \\ s\_a(s\_t(color\_inkjet\_queue)) \prec use(color\_inkjet) \end{array} \right\}$$

where s_a means "*select action*" and s_t means "*send to*". Once the user requirements are provided and added to the set Π, the agent may reason about which intention is warranted and it could then select the appropriate action. For instance, consider a situation in which the agent's perception function returns the fact that there is a file queued for printing. Then, a new knowledge base Π' will be produced by means of the belief revision mechanism (as will be described below), defined as $\Pi' = \Pi \cup \{file\_for\_printing\}$. From ($\Pi'$,$\Delta$) there exists a warrant for the intention $use(inkjet)$ because there is no counter-argument for $\mathcal{A}$= $\{use(inkjet) \prec file\_for\_printing\}$.

Suppose that, at a later time, the printing of another file is required with the "color" constraint, *i.e.*, $\Pi' = \Pi \cup \{file\_for\_printing, color\}$. Now, from ($\Pi'$,$\Delta$) it is not possible to find a warrant for $use(inkjet)$ because argument $\mathcal{A}$ is defeated by argument $\mathcal{A}_2$=$\{use(color\_inkjet) \prec file\_for\_printing, color \}$ and there is a warrant for the intention $use(color\_inkjet)$. Therefore, the agent intention will change. Finally, observe that once an intention is warranted, a warrant for the appropriate action may be obtained using the last three rules of Δ.

Next, we will consider the problem of belief dynamics in rational agents. Given that our aim is to use DeLP for knowledge representation and reasoning, we will consider a non-prioritized mechanism capable of revising the epistemic state of such agent that will be represented as a program $\mathcal{P}$=(Π, Δ).

Belief Revision Mechanism

In this section, we will develop a formalism for the updating of the beliefs of an agent's knowledge base (or belief base). This knowledge base will be represented as a defeasible logic program of the form $\mathcal{P}$=(Π, Δ). The revision operator will modify the set Π in a way that follows the one presented in (Falappa, Kern-Isberner, & Simari 2002). However, it will aim to preserve in Δ, after transforming it, the information revised from Π.

Belief Revision systems are logical frameworks for modeling the dynamics of knowledge, that is, the way an agent modifies its beliefs when it receives new information. The main problem arises when the information received is inconsistent with the set of beliefs that represents the agent's epistemic state. For instance, suppose the agent believes that "*all metals are solid*" and then it finds out that "*mercury is a metal in liquid state under normal pressure.*" Certainly, the agent needs to revise its beliefs in order to accept the new information while preserving as much of the old information as possible and maintain the consistency of its knowledge base. Our formalism is designed to handle the problem of changing the beliefs that the agent has about a particular state of the world as *new* information arrives (usually called *revision*). However, it is not designed to deal with *changes* in the world (usually called *updates*).

One of the most discussed properties in the literature dealing with revision operators is the property of *success.* This property establishes that the new information has primacy over the beliefs of an agent. Here, we will use a non-prioritized revision operator (Falappa, Kern-Isberner, & Simari 2002), *i.e.*, new information has no precedence over the agent's current beliefs. In this operator, new information will be supported by an *explanation.* Every explanation contains an *explanans* (the beliefs that support a conclusion) and an *explanandum* (the final conclusion supported by the explanans). Thus, an explanation, to be defined below, is a set of sentences with some restrictions. The intention behind this development is that the beliefs which will be deleted from the belief base Π could be preserved in an alternate set, changing their epistemic status to defeasible knowledge.

Epistemic Model and Construction

Every rational agent working in a dynamic environment must have a way to model *change*. When we talk about change, we assume that there is an object of the language

that changes, that is, the *belief state* (Hansson 1999). Since there are various possible constructions to be considered as models of the belief state, we must define the *epistemic model*, that is, the way in which the belief states are represented.

There are different ways to represent belief states. We may use *belief sets*, that is, sets of sentences closed under logical consequence, *i.e.*, everything that follows logically from a belief set is an element of it. This alternative for the representation of belief states has some advantages on the *Knowledge Level* (Newell 1982), but is not computationally adequate.

Another alternative for the representation of belief states as belief sets is the use of *belief bases*, that is, sets of sentences that are not logically closed (Hansson 1997). A belief base contains information about the justificatory structure of the respective belief state, and this structure is closely connected to the dynamic behavior of the belief system. Two belief bases with the same logical closure can behave differently under operations of change, that is, they can (dynamically) represent different ways of holding the same belief state. That is, some of our beliefs have no independent standing, but they arise as inference from our more basic beliefs, on which they are entirely contingent (Hansson 1997).

Every time an agent needs to incorporate some belief α to his/her belief base, it must perform two major tasks: (1) add the new belief α to the belief base and (2) ensure that the resulting belief base is consistent. The *Levi Identity* (Levi 1977) provides the traditional way of adding beliefs: first contracting by $\sim\alpha$, eliminating all possible derivation of $\sim\alpha$ from the belief base, and then adding α to the resulting belief base. This type of operator is *internal* because the sub-operation of contraction takes place inside the original belief base. Alternatively, the two sub-operations may be effected in reverse order (Hansson 1993): adding α and then eliminating $\sim\alpha$ from the resulting belief base. This kind of operator is *external* because the contraction takes place outside of the original belief base. It is important to note that consistency is preserved on every step of internal revision, whereas in external revision the intermediate belief base will often be inconsistent. In these two operators the new information is always accepted, and it is for this reason that they are called *prioritized*.

The next modification of external revision leads to the *semi-revision operator* (Hansson 1997): adding α to the belief base and then eliminating all possible inconsistency from the resulting belief base. If we replace the single sentence α by a set of sentences A, we get the *Operator of Revision by a Set of Sentences* (Falappa, Kern-Isberner, & Simari 2002). These two operators are *non-prioritized* because the new information could be totally or partially rejected and they typically involve the temporary acceptance of an inconsistent set.

Revision by a Set of Sentences

As we said before, the knowledge base will be represented as a defeasible logic program of the form $\mathcal{P}$=(Π, Δ). In this section, we will consider the revision of the strict part of the knowledge base, Π, as the knowledge changes.

The *Revision Operator by a Set of Sentences* will be a function that takes two sets of sentences and produces a new set of sentences. In this section, when we refer to *sentences*, these will be facts or strict rules, and a belief base will be a finite set of sentences Π which is not closed under logical consequence.

There are two standard ways of defining an operator of revision by a set of sentences: kernel mode and partial meet mode. The first one uses an *external incision function* and the second one uses an *equitable selection function* (Falappa, Kern-Isberner, & Simari 2002). The first construction is based on the concept of kernel set (Hansson 1994).

In the definitions below we will use the language $\mathcal{L}$ defined by the elements of $\mathcal{P} = (\Pi, \Delta)$. Notice that in Defeasible Logic Programming "$\prec$" and "$\leftarrow$" are meta-linguistic symbols.

Definition 1 *Let Π be a set of sentences and α a sentence. Then $\Pi^{\perp\!\!\!\perp}\alpha$ is the set of all sets Π' if and only if the following conditions are satisfied:*

1. **Derivability:** $\Pi' \subseteq \Pi$, $\Pi' \vdash \alpha$, *and*
2. **Minimality:** *for all proper subsets Π'' of Π', $\Pi'' \nvdash \alpha$.*

The set $\Pi^{\perp\!\!\!\perp}\alpha$ is called the kernel set, *and its elements are called the* α-kernels of Π.

Example 1 *If $\Pi = \{p, q, (r \leftarrow p, q), t, (r \leftarrow t), u, (v \leftarrow u)\}$ then $\Pi^{\perp\!\!\!\perp}r = \{\{p, q, (r \leftarrow p, q)\}, \{t, (r \leftarrow t)\}\}$, $\Pi^{\perp\!\!\!\perp}w = \{\}$ because $\Pi \nvdash w$, and $\Pi^{\perp\!\!\!\perp}(r \leftarrow r) = \{\{\}\}$ because $r \leftarrow r$ is a tautology.*

Definition 2 *Let Π be a set of sentences. An* external incision function for Π *is a function $\sigma : \mathbf{2}^{\mathbf{2}^{\mathcal{L}}} \Rightarrow \mathbf{2}^{\mathcal{L}}$, such that for any set $A \subseteq \mathcal{L}$, the following conditions hold:*

1. $\sigma((\Pi \cup A)^{\perp\!\!\!\perp}\bot) \subseteq \cup((\Pi \cup A)^{\perp\!\!\!\perp}\bot)$
2. *If $X \in (\Pi \cup A)^{\perp\!\!\!\perp}\bot$ and $X \neq \varnothing$ then $(X \cap \sigma((\Pi \cup A)^{\perp\!\!\!\perp}\bot)) \neq \varnothing$*

For the limit case in which $(\Pi \cup A)^{\perp\!\!\!\perp}\bot = \varnothing$, σ is defined as $\varnothing$, i.e., $\sigma((\Pi \cup A)^{\perp\!\!\!\perp}\bot) = \varnothing$.

Definition 3 *Let Π and A be sets of sentences, and "σ" an external incision function for Π. The operator $\circ : \mathbf{2}^{\mathcal{L}} \times \mathbf{2}^{\mathcal{L}} \Rightarrow \mathbf{2}^{\mathcal{L}}$ of* kernel revision by a set of sentences, *is defined as $\Pi \circ A = (\Pi \cup A) \setminus \sigma((\Pi \cup A)^{\perp\!\!\!\perp}\bot)$.*

This operator works by adding A to Π and then eliminating all possible inconsistency from the resulting set. This is done by means of an incision function "σ" that makes a "cut" on each minimally inconsistent subset of $\Pi \cup A$.

Example 2 *Let $\Pi = \{p, (r \leftarrow p), t, (u \leftarrow t)\}$ and $A = \{p, q, (\sim r \leftarrow p, q)\}$. Then $(\Pi \cup A)^{\perp\!\!\!\perp}\bot$ is equal to $\{\{p, q, (r \leftarrow p), (\sim r \leftarrow p, q)\}\}$. The incision function σ has to make a "cut" on the one minimal inconsistent subset of $\Pi \cup A$: $\{p, q, (r \leftarrow p), (\sim r \leftarrow p, q)\}$. Some possible incision functions, described by its images, are $\sigma_1 : \{p\}$, $\sigma_2 : \{q\}$, $\sigma_3 : \{(r \leftarrow p)\}$, $\sigma_4 : \{p, (r \leftarrow p)\}$,* etc. *Their respective associated revisions by a set of sentences ($\Pi \circ A$) are:*

1. $\{q, (r \leftarrow p), t, (u \leftarrow t), (\sim r \leftarrow p, q)\}$ *for σ_1,*
2. $\{p, (r \leftarrow p), t, (u \leftarrow t), (\sim r \leftarrow p, q)\}$ *for σ_2,*

3. $\{p, q, t, (u \leftarrow t), (\sim r \leftarrow p, q)\}$ *for* σ_3,
4. $\{q, t, (u \leftarrow t), (\sim r \leftarrow p, q)\}$, *for* σ_4, etc.

Now, we will present a second construction of the operator of revision by a set of sentences based on the concept of remainder set (Alchourrón, Gärdenfors, & Makinson 1985).

Definition 4 *Let* Π *be a set of sentences and* α *a sentence. Then* $\Pi^{\perp}\alpha$ *is the set of all subsets* Π' *of* Π *such that the following conditions are satisfied:*

1. **Non-derivability:** $\Pi' \subseteq \Pi$ *and* $\Pi' \nvdash \alpha$, *and*
2. **Maximality:** *if* Π'' *is a subset of* Π *such that* $\Pi' \subset \Pi'' \subseteq \Pi$ *then* $\Pi'' \vdash \alpha$.

The set $\Pi^{\perp}\alpha$ *is called the* remainder set *of* Π *with respect to* α, *and its elements are called the* α-remainders *of* Π.

Example 3 *If* $\Pi = \{p, (r \leftarrow p), t, (r \leftarrow t), u, (v \leftarrow u)\}$ *then* $\Pi^{\perp}r = \{\{p, t, u, (v \leftarrow u)\}, \{p, (r \leftarrow t), u, (v \leftarrow u)\}, \{(r \leftarrow p), t, u, (v \leftarrow u)\}, \{(r \leftarrow p), (r \leftarrow t), u, (v \leftarrow u)\}\}$, $\Pi^{\perp}w = \Pi$ *because* $\Pi \nvdash w$, *and* $\Pi^{\perp}(r \leftarrow r) = \{\}$ *because* $r \leftarrow r$ *is a tautology.*

Definition 5 *Let* Π *be a set of sentences. An* external selection function for Π *is a function* $\gamma : \mathbf{2}^{\mathbf{2}^{\mathcal{L}}} \Rightarrow \mathbf{2}^{\mathbf{2}^{\mathcal{L}}}$, *such that for any set* $A \subseteq \mathcal{L}$, *it holds that:*

1. $\gamma((\Pi \cup A)^{\perp}\perp) \subseteq (\Pi \cup A)^{\perp}\perp$
2. $\gamma((\Pi \cup A)^{\perp}\perp) \neq \varnothing$

Example 4 *Given* $\Pi = \{p, q, r\}$ *and* $A = \{\sim p, \sim q\}$ *then* $\Pi \cup A = \{p, q, r, \sim p, \sim q\}$ *and* $(\Pi \cup A)^{\perp}\perp = \{\{p, q, r\}, \{\sim p, q, r\}, \{p, \sim q, r\}, \{\sim p, \sim q, r\}\}$. *Consequently, some possible results of* $\gamma((\Pi \cup A)^{\perp}\perp)$ *are* $\{\{\sim p, \sim q, r\}\}$ *or* $\{\{\sim p, q, r\}\}$ *or* $\{\{p, q, r\}, \{\sim p, q, r\}\}$ *or* $\{\{p, q, r\}, \{\sim p, q, r\}, \{p, \sim q, r\}\}$.

Definition 6 *Let* Π *be a set of sentences and* γ *an external selection function for* Π. *Then* γ *is an* equitable selection function for Π *if* $(\Pi \cup A)^{\perp\!\!\perp}\perp = (\Pi \cup B)^{\perp\!\!\perp}\perp$ *implies that* $(\Pi \cup A) \setminus \cap\gamma((\Pi \cup A)^{\perp}\perp) = (\Pi \cup B) \setminus \cap\gamma((\Pi \cup B)^{\perp}\perp)$.

The intuition behind this definition is that, if the set of minimally inconsistent subsets of $\Pi \cup A$ is equal to the set of minimally inconsistent subsets of $\Pi \cup B$, then α is erased in the selection of $\perp$-remainders of $\Pi \cup A$ if and only if it is erased in the selection of $\perp$-remainders of $\Pi \cup B$.

Definition 7 *Let* Π *and* A *be sets of sentences and "*γ*" an equitable selection function for* Π. *The operator* $\circ : \mathbf{2}^{\mathcal{L}} \times \mathbf{2}^{\mathcal{L}} \Rightarrow \mathbf{2}^{\mathcal{L}}$ *of* partial meet revision by a set of sentences *is defined as* $\Pi \circ A = \cap\gamma((\Pi \cup A)^{\perp}\perp)$.

The mechanism used by this operator is to add A to Π and then eliminate from the result all possible inconsistencies by means of an equitable selection function that makes a choice among the maximally consistent subsets of $\Pi \cup A$ and intersect them. An axiomatic characterization for operators of kernel and partial meet revision by sets of sentences can be founded in (Falappa, Kern-Isberner, & Simari 2002).

These operators can be used by an agent to update its knowledge. For instance, suppose that in a revision process, we eliminate a conditional sentence of the form $\beta(X) \leftarrow \alpha(X)$. This sentence ensures that any object X satisfying the predicate α is an object satisfying the predicate β. Suppose now that the agent receives new information which establishes that there could be exceptional objects that satisfy α but do not satisfy β, *i.e.*, satisfy $\sim\beta$. In this case, we could discard the original rule $\beta(X) \leftarrow \alpha(X)$ because we have accepted that the rule has exceptions that make it wrong. However, this policy, although safe and correct, is somewhat inadequate because it produces a complete loss of information.

Here, we will propose an alternative that preserves a different form of this type of sentences that weakens the relation between the head and the body of the rule. Thus, the strong form of the rule will be transformed into two defeasible rules. In the particular case of the example above, the agent could preserve $\beta(X) \prec\!\!\!- \alpha(X)$ in order to keep some of the inferential power that it had before. The idea is that some defeasible rule of the form $\beta(X) \prec\!\!\!- \alpha(X)$ in Δ is the transformation of some rule $\beta(X) \leftarrow \alpha(X)$ previously included in the strong knowledge but eliminated by some change operator. Instead of completely eliminating this sentence, we propose to preserve a syntactic transformation of it in a different set, as defined below.

Definition 8 *Let* $\mathcal{P} = (\Pi, \Delta)$ *be a defeasible logic program. Let* $\delta = \beta \leftarrow \alpha$ *be a strict rule in* Π. *A* positive transformation *of* δ, *noted by* $T^{+}(\delta)$, *is a sentence of the form* $\beta \prec\!\!\!- \alpha$; *a* negative transformation *of* δ, *noted by* $T^{-}(\delta)$, *is a sentence of the form* $\sim\alpha \prec\!\!\!- \sim\beta$.

Definition 9 *Let* $\mathcal{P} = (\Pi, \Delta)$ *be a defeasible logic program, "*$\circ$*" an operator of revision by a set of sentences for* Π *and* A *a set of sentences. The* composed revision *of* (Π, Δ) *with respect to* A *is defined as* $(\Pi, \Delta) \star A = (\Pi', \Delta')$ *such that* $\Pi' = \Pi \circ A$ *and* $\Delta' = \Delta \cup \Delta''$ *where* $\Delta'' = \{T^{+}(\alpha) : \alpha \in (\Pi \setminus \Pi \circ A)\} \cup \{T^{-}(\alpha) : \alpha \in (\Pi \setminus \Pi \circ A)\}$.

The set Π' contains the revised non-defeasible beliefs, while Δ'' contains the general beliefs eliminated from Π and transformed into defeasible rules.

Example 5 *Suppose that we have an agent with the intentions of discovering precise properties about metals. Suppose that the beliefs of that agent are represented by a defeasible logic program* $\mathcal{P} = (\Pi, \Delta)$ *where:*

$$\Pi = \left\{ \begin{array}{l} metal(fe) \\ metal(hg) \\ solid(X) \leftarrow metal(X) \\ \sim liquid(X) \leftarrow solid(X) \\ \sim solid(X) \leftarrow liquid(X) \end{array} \right\} \text{ and } \Delta = \{\}$$

where hg *means "*mercury*" and* fe *means "*iron*". Then, it receives the following explanation* A *for* $liquid(hg)$*:*

$$\{(liquid(hg) \leftarrow metal(hg), pressure(normal))\} \cup \{metal(hg), pressure(normal)\}$$

Based on the kernel revision by a set of sentences, it is necessary to remove any inconsistency from the following sets:

$$\Pi_1 = \left\{ \begin{array}{l} metal(hg) \\ pressure(normal) \\ solid(X) \leftarrow metal(X) \\ liquid(hg) \leftarrow metal(hg), pressure(normal) \\ \sim liquid(X) \leftarrow solid(X) \end{array} \right\}$$

$$\Pi_2 = \left\{ \begin{array}{l} metal(hg) \\ pressure(normal) \\ solid(X) \leftarrow metal(X) \\ liquid(hg) \leftarrow metal(hg), pressure(normal) \\ \sim solid(X) \leftarrow liquid(X) \end{array} \right\}$$

Π_1 *and* Π_2 *represent the* $\perp$*-kernels (minimally inconsistent subsets) of* $\Pi \cup A$*, that is,* $(\Pi \cup A)^{\perp\!\!\!\perp} \perp = \{\Pi_1, \Pi_2\}$*. Depending on the preference criteria among the sentences of* Π_1 *and* Π_2*, a possible result of* $(\Pi, \Delta) \star A = (\Pi', \Delta')$ *is:*

$$\Pi' = \left\{ \begin{array}{l} metal(fe) \\ metal(hg) \\ pressure(normal) \\ liquid(hg) \leftarrow metal(hg), pressure(normal) \\ \sim liquid(X) \leftarrow solid(X) \\ \sim solid(X) \leftarrow liquid(X) \end{array} \right\}$$

$$\Delta' = \{solid(X) \prec metal(X), \sim metal(X) \prec \sim solid(X)\}$$

In this way defeasible conditionals can be generated using the revision operator and it extends the inference power of an agent based on the BDI model.

Conclusions and future work

In this work, we have introduced a *non-prioritized* belief revision operator, the *Revision Operator by a Set of Sentences*, for changing the agent's beliefs. This operator has the desirable property of conserving the greatest amount of information possible by transforming strict rules into defeasible ones. These beliefs are represented in a Knowledge Base using the language of Defeasible Logic Programming. The combination of both frameworks results in a formalism for knowledge representation and reasoning about beliefs and intentions. For the next step, we will explore the properties of this operator and develop multi-agent applications. Since the DeLP system is currently implemented, we will extend this implementation to include the new operator.

References

Alchourrón, C.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic* 50:510–530.

Bratman, M. 1987. *Intentions, Plans, and Practical Reasoning*. Cambridge, Massachusetts: Harvard University Press (Reprinted by CSLI Publications.

Falappa, M. A.; Kern-Isberner, G.; and Simari, G. R. 2002. Explanations, belief revision and defeasible reasoning. *Artificial Intelligence Journal* 141(1-2):1–28.

García, A. J., and Simari, G. R. 1999. Parallel defeasible argumentation. *Journal of Computer Science and Technology Special Issue: Artificial Intelligence and Evolutive Computation. http://journal.info.unlp.edu.ar/* 1(2):45–57.

García, A. J., and Simari, G. R. 2004. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming* 4(1):95–138.

Georgeff, M.; Pell, B.; Pollack, M.; Tambe, M.; and Wooldridge, M. 1999. The belief-desire-intention model of agency. In Müller, J.; Singh, M. P.; and Rao, A. S., eds., *Proceedings of the 5th International Workshop on Intelligent Agents V: Agent Theories, Architectures, and Languages (ATAL-98)*, volume 1555, 1–10. Springer-Verlag: Heidelberg, Germany.

Hansson, S. O. 1993. *Reversing the Levi Identity. The Journal of Philosophical Logic* 22:637–669.

Hansson, S. O. 1994. Kernel contraction. *The Journal of Symbolic Logic* 59:845–859.

Hansson, S. O. 1997. *Semi-Revision. Journal of Applied Non-Classical Logic* 7:151–175.

Hansson, S. O. 1999. *A Textbook of Belief Dymanics: Theory Change and Database Updating.* Kluwer Academic Publishers.

Huhns, M. N., and Singh, M. P. 1997. *Readings in Agents*. Morgan Kaufmann Pub.

Jennings, N. R., and Wooldridge, M. J., eds. 1998. *Agent Technology: Foundations, Applications, and Markets*. Berlin: Springer-Verlag.

Levi, I. 1977. *Subjunctives, Dispositions and Chances. Synthese* 34:423–455.

McCarthy, J., and Hayes, P. J. 1969. Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B., and Michie, D., eds., *Machine Intelligence 4*. Edinburgh University Press. 463–502. reprinted in McC90.

McCarthy, J. 1958. Programs with common sense. In *Teddington Conference on the Mechanization of Thought Processes*.

McCarthy, J. 1990. *Formalization of common sense, papers by John McCarthy edited by V. Lifschitz*. Ablex.

Newell, A. 1982. *The Knowledge Level. Artificial Intelligence* 18:87–127.

Nute, D. 1994. Defeasible logic. In Gabbay, D.; Hogger, C.; and J.A.Robinson., eds., *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol 3*. Oxford University Press. 355–395.

Prakken, H., and Vreeswijk, G. 2000. Logical systems for defeasible argumentation. In D.Gabbay., ed., *Handbook of Philosophical Logic, 2nd ed.* Kluwer Academic Pub.

Rao, A. S., and Georgeff, M. P. 1991. Modeling rational agents within a BDI-architecture. In Allen, J. F.; Fikes, R.; and Sandewall, E., eds., *KR'91: Principles of Knowledge Representation and Reasoning*. San Mateo, California: Morgan Kaufmann. 473–484.

Rao, A. S., and Georgeff, M. P. 1992. An abstract architecture for rational agents. In Rich, C.; Swartout, W.; and Nebel, B., eds., *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR&R'92)*, 439–449. Cambridge, MA, USA: Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

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

Simari, G. R.; Chesñevar, C. I.; and García, A. J. 1994. The role of dialectics in defeasible argumentation. In *XIV International Conference of the Chilenean Computer Science Society*.

Weiss, G., ed. 1999. *Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence*. Cambridge, Massachussetts: The MIT Press.

Wooldridge, M., and Jennings, N. 1994. Agent theories, architectures, and languages: A survey. In *Intelligent Agents - Theories, Architectures, and Languages*, volume 890. Michael Wooldridge and Nicholas R. Jennings (ed.) Springer-Verlag. 1–32.

Wooldridge, M., and Rao, A. 1999. *Foundations of Rational Agency*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Wooldridge, M. 2000. *Reasoning about Rational Agents*. The MIT Press: Cambridge, MA, USA.