5.8 Defeasible Reasoning about Beliefs and Desires

Defeasible Reasoning About Beliefs and Desires

Nicolás D. Rotstein and **Alejandro J. García**
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Department of Computer Science and Engineering, Universidad Nacional del Sur,
Email: {ndr,ajg}@cs.uns.edu.ar

Abstract

In this paper we show how a deliberative agent can represent beliefs and desires and perform defeasible reasoning in order to support its derived beliefs. Strict and defeasible filtering rules can be used by the agent for selecting among its desires a proper one that fits in the particular situation it is involved. Thus, defeasible argumentation will be used for reasoning about desires. Application examples from a robotic soccer domain will be given.

Introduction

This article addresses the problem of having a deliberative intelligent agent built upon an architecture that relies on sets of Beliefs and Desires (*e.g.*, the BDI architecture (Bratman, Israel, & Pollack 1991; Rao & Georgeff 1995; Rao 1996)). The proposed approach allows the agent to perform defeasible reasoning in order to support its derived beliefs and offers a defeasible argumentation framework for selecting desires.

In our approach, the agent will represent information that it perceives directly from its environment, and in addition to these *perceived beliefs*, the agent may represent other knowledge in the form of strict and defeasible rules. Then, using a defeasible argumentation formalism it will be able to to obtain a *warrant* for its *derived beliefs*.

We will introduce a reasoning formalism for selecting those desires that are suitable to be carried out by the agent. In order to perform this selection, the agent will use its beliefs (that represent the current situation) and a defeasible logic program composed by *filtering rules*.

Warranting agent's beliefs and perception

In this approach, agent's beliefs will correspond to the semantics[1] of a *defeasible logic program* $\mathcal{P}_\mathsf{B}$ = (Π_B,Δ_B). In this kind of programs (García & Simari 2004) two different sets are distinguished: the set Δ_B for representing tentative knowledge in the form of defeasible rules; and the set Π_B for representing non-tentative, sound knowledge in the form of strict rules and facts. The information that the agent perceives directly from its environment is represented in Π_B with a subset of facts denoted Φ. Thus, in the set Π_B two disjoint subsets will be distinguished: the subset Φ of *perceived beliefs* that will be updated dynamically, and a subset Π formed with strict rules and facts that will represent static knowledge. Therefore, Π_B= $\Phi \cup \Pi$. As we will explain below, in order for program $\mathcal{P}_\mathsf{B}$ to behave correctly, some restrictions over Π_B will be imposed.

In addition to the perceived beliefs, the agent may use strict and defeasible rules from $\mathcal{P}_\mathsf{B}$ in order to obtain a *warrant* for its *derived beliefs* (see Definition 1). A brief explanation of how warrants are obtained using the defeasible argumentation formalism of DeLP, and the definitions of facts, strict and defeasible rules are included below (the interested reader is referred to (García & Simari 2004) for a detailed explanation).

Representing knowledge and reasoning with DeLP

In DeLP, knowledge is represented using facts, strict rules or defeasible rules:

- *Facts* are ground literals representing atomic information or the negation of atomic information using the strong negation "$\sim$" (*e.g.*, $hasBall(opponent)$).
- *Strict Rules* are denoted $L_0 \leftarrow L_1, \ldots, L_n$, where the *head* L_0 is a ground literal and the *body* $\{L_i\}_{i>0}$ is a set of ground literals. (*e.g.*, $\sim hasBall(myTeam) \leftarrow hasBall(opponent)$).
- *Defeasible Rules* are denoted $L_0 \prec L_1, \ldots, L_n$, where the *head* L_0 is a ground literal and the *body* $\{L_i\}_{i>0}$ is a set of ground literals. (*e.g.*, $\sim pass(mate1) \prec marked(mate1)$).

Syntactically, the symbol "$\prec$" is all that distinguishes a defeasible rule from a strict one. Pragmatically, a defeasible rule is used to represent defeasible knowledge, *i.e.*, tentative information that may be used if nothing could be posed against it. A defeasible rule "$Head \prec Body$" is understood as expressing that "*reasons to believe in the antecedent* $Body$ *provide reasons to believe in the consequent* $Head$" (Simari & Loui 1992).

A Defeasible Logic Program $\mathcal{P}$ is a set of facts, strict rules and defeasible rules. When required, $\mathcal{P}$ is denoted (Π, Δ) distinguishing the subset Π of facts and strict rules, and the subset Δ of defeasible rules. Observe that strict

[1] Since the semantics of DeLP is skeptical, there is only one.

and defeasible rules are ground. However, following the usual convention (Lifschitz 1996), some examples will use "schematic rules" with variables. Given a "schematic rule" R, $Ground(R)$ stands for the set of all ground instances of R. Given a program $\mathcal{P}$ with schematic rules, we define (Lifschitz 1996): $Ground(\mathcal{P}) = \bigcup_{R \in \mathcal{P}} Ground(R)$. In order to distinguish variables, they are denoted with an initial uppercase letter.

Strong negation is allowed in the head of program rules, and hence may be used to represent contradictory knowledge. From a program (Π, Δ) contradictory literals could be derived, however, the set Π (which is used to represent non-defeasible information) must possess certain internal coherence. Therefore, Π has to be non-contradictory, *i.e.*, no pair of contradictory literals can be derived from Π. Given a literal L the complement with respect to strong negation will be denoted $\overline{L}$ (*i.e.*, $\overline{a}$=$\sim a$ and $\overline{\sim a}$=a).

DeLP incorporates an argumentation formalism for the treatment of the contradictory knowledge that can be derived from (Π, Δ). This formalism allows the identification of the pieces of knowledge that are in contradiction. A dialectical process is used for deciding which information prevails. In particular, the argumentation-based definition of the inference relation makes it possible to incorporate a treatment of preferences in an elegant way.

In DeLP a literal L is *warranted* from (Π, Δ) if there exists a non-defeated argument $\mathcal{A}$ supporting L. In short, an *argument* for a literal L, denoted $\langle \mathcal{A}, L \rangle$, is a minimal set of defeasible rules $\mathcal{A} \subseteq \Delta$, such that $\mathcal{A} \cup \Pi$ is non-contradictory and there is a derivation for L from $\mathcal{A} \cup \Pi$. In order to establish if $\langle \mathcal{A}, L \rangle$ 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 *argumentation line* is constructed, where each argument defeats its predecessor in the line (for a detailed explanation of this dialectical process see (García & Simari 2004)).

In DeLP, given a query Q there are four possible answers[2]: YES, if Q is warranted; *no*, if the complement of Q is warranted; *undecided*, if neither Q nor its complement is warranted; and *unknown*, if Q is not in the language of the program.

Perceived and Warranted Beliefs

As stated above, agent's beliefs will correspond to the semantics of a defeasible logic program (Π_{B},Δ_{B}), and the set Π_{B} will represent perceived and static information (Π_{B}= $\Phi \cup \Pi$). Since Π_{B} has to be non-contradictory, some restrictions about perception are imposed:

1. We assume that perception is correct in the sense that it will not give a pair of contradictory literals. Whenever this happens both literals will be ignored.
2. We will also require that no perceived literal in Φ can be derived directly from Π.

Thus, if Π is non-contradictory and these two restrictions are satisfied, then Π_{B} will also be non-contradictory. The next definition introduces the different types of belief that an agent will obtain from a defeasible logic program (Π_{B},Δ_{B}).

Definition 1 (belief types) *A **perceived belief** is a fact in Φ that represents information that the agent has perceived directly from its environment. A **strict belief** is a literal that is not a perceived belief, and it is derived from $\Pi_{\mathsf{B}} = \Pi \cup \Phi$ (i.e., no defeasible rules are used for its derivation). A **defeasible belief** is a warranted literal L supported by an argument $\langle \mathcal{A}, L \rangle$ that uses at least one defeasible rule (i.e., $\mathcal{A} \neq \emptyset$). Finally, a **derived belief** is a strict or a defeasible belief. We will denote with B_s the set of strict beliefs, and with B_d the set of defeasible beliefs. Therefore, in any given situation the beliefs of an agent will be $\mathsf{B} = \Phi \cup \mathsf{B}_s \cup \mathsf{B}_d$.*

Observe that the sets Φ, B_s and B_d are disjoint sets. Observe also that, although perceived beliefs are facts in Π_{B}, there can be other facts in Π_{B} that are not perceived. For instance, facts that represent agent's features, roles, etc. These facts that do not represent perceived information are persistent in the sense that they will not change with perception. For example: $myRole(defender)$, $opponent(o1)$.

In this approach we assume a perception function that provides the agent with information about its environment. This function will be invoked by the agent in order to update its perceived beliefs set Φ. When this happens the new information obtained must override the old one following some criterion. Updating a set of literals is a well-known problem and many solutions exist in the literature.

Example 1 *Consider an agent Ag that has the following program (Π_{B},Δ_{B}). Here, the set Π_{B} was divided distinguishing the subset Φ of perceived facts, and the subset Π of non-perceived information.*

$$\Phi = \left\{ \begin{array}{l} hasBall(t1) \\ marked(t1) \end{array} \right\}$$

$$\Pi = \left\{ \begin{array}{l} mate(t1) \\ opponent(o1) \\ \sim mate(X) \leftarrow opponent(X) \\ \sim receive(self) \leftarrow hasBall(self) \end{array} \right\}$$

$$\Delta_{\mathsf{B}} = \left\{ \begin{array}{l} receive(self) \prec hasBall(X), mate(X) \\ \sim receive(self) \prec marked(self) \\ \sim receive(self) \prec hasBall(X), \sim mate(X) \end{array} \right\}$$

In this example, Ag has the following perceived beliefs: $hasBall(t1)$ (the player $t1$ has the ball), and $marked(t1)$ (its teammate $t1$ is marked). Besides its perceived beliefs, it has two other facts that are strict beliefs: $mate(t1)$ ($t1$ is a teammate) and $opponent(o1)$ ($o1$ is an opponent). The set Π has also two strict rules representing that "an opponent is not a teammate" and that "Ag cannot receive the ball from itself". Observe that it can also infer the strict belief: $\sim mate(o1)$ (o1 is not a teammate).

The set of defeasible rules Δ_{B} represents that: "if a teammate has the ball, then Ag may receive a pass

[2]The implementation (interpreter) of DeLP that satisfies the semantics described in (García & Simari 2004) is currently accessible online at `http://lidia.cs.uns.edu.ar/DeLP`.

from it", "being marked is a good reason for not receiving a pass", and "if the one that has the ball is not a teammate, then there are good reasons for not receiving the ball from it". Thus, from (Π_{B},Δ_{B})*, the argument* $\{receive(self) \prec hasBall(t1), mate(t1)\}$ *has no defeaters, and therefore, there is a warrant for one defeasible belief:* $receive(self)$ *(Ag may receive a pass).*

Consider now that, upon perception, the set Φ *is updated to* Φ= $\{hasBall(t1), marked(t1), marked(self)\}$ *(i.e., the original situation has changed only in that the agent is now being marked); then, from this new program, the argument for* $receive(self)$ *has a "blocking defeater", what means that the DeLP answer for* $receive(self)$ *and* $\sim receive(self)$ *will be both* UNDECIDED*. Figure 1 shows this situation, where two incomparable arguments block each other. There, arguments are depicted as triangles containing the defeasible rules that form them. The double arrow represents that both arguments attack each other with equal strength.*

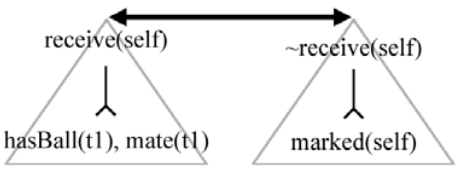

Figure 1: *Undecided situation due to blocking defeaters.*

Consider a new situation where Φ= $\{hasBall(o1)\}$*. Here, the DeLP answer for* $receive(self)$ *is* NO*, because there is a warrant for* $\sim receive(self)$ *supported by the non-defeated argument:*

$$\{\sim receive(self) \prec hasBall(o1), \sim mate(o1)\}.$$

A preference criterion between contradictory arguments is needed in order to prevent blocking situations. For a deeper discussion on this matter we refer to (Chesñevar, Maguitman, & Loui 2000; Prakken & Vreeswijk 2002).

The following propositions show that, although $\mathcal{P}_{\mathsf{B}}$ represents contradictory information, the sets of beliefs B of an agent will be non-contradictory.

Proposition 1 *The set of beliefs* B *of an agent is a set of warranted literals.*
Proof: As proved in (García & Simari 2004), empty arguments have no defeaters and therefore, if a literal can be derived from Π_{B} *then it is warranted. Thus, perceived beliefs are warranted literals, because a fact has an empty argument that supports it. Strict beliefs are also warranted because they are derived using only strict rules and facts from* Π_{B} *and therefore, supported by an empty argument. Defeasible beliefs are warranted by definition.*

Proposition 2 *The set of beliefs* B *of an agent is a non-contradictory set.*
Proof: In order to be a contradictory set, it should have two complementary beliefs (L *and* $\sim L$*). However, proposition 1 states that* B *is a set of warranted literals, and from a defeasible logic program it is not possible to obtain a warrant for a literal* L *and also a warrant for* $\sim L$*.*

Desires Filtering and Selection

In our approach, agents desires will be represented by a set of literals D. This set will contain a literal for representing each desire that the agent might want to achieve, along with its complement; that is, if $L \in \mathsf{D}$, then $\overline{L} \in \mathsf{D}$. As we will explain in detail below, we will assume that beliefs and desires are represented with separate names, *i.e.*, $\mathsf{D} \cap \mathsf{B} = \emptyset$, (see Remark 1).

Example 2 *According to our application domain, the set* D *of all possible desires for a robotic soccer agent could be:*

$$\mathsf{D} = \left\{ \begin{array}{ll} shoot & \sim shoot \\ pass(Mate) & \sim pass(Mate) \\ carry & \sim carry \\ mark(Opp) & \sim mark(Opp) \\ goto(Place) & \sim goto(Place) \end{array} \right\}$$

That is, the set of possible desires of the agent includes: to shoot on goal, to pass the ball to a teammate, to carry the ball, to mark an opponent, to go to a specific position in the field, and each corresponding complement.

The set D represents all the desires that the agent may want to achieve. However, depending on the situation in which it is involved, there could be some desires that will be impossible to be carried out. For example, consider a situation in which the agent does not have the ball and the ball is in a place p, then, the desire $shoot$ will not be possible to be carried out, whereas $goto(p)$ could be a possible option.

Therefore, agents should reason about its desires in order to select the appropriate ones. Following the spirit of the BDI model, once appropriate desires are detected, the agent may select (and commit to) a specific intention (goal), and then select appropriate actions to fulfill that intention.

In this section, we will introduce a reasoning formalism for selecting from D those desires that are suitable to be carried out. In order to perform this selection, the agent will use its beliefs (representing the current situation) and a defeasible logic program (Π_F,Δ_F) composed by *filtering rules* as defined below.

Definition 2 (Filtering rule) *Let* D *be the set of desires of an agent, a* filtering rule *is a strict or defeasible rule that has a literal* $L \in \mathsf{D}$ *in its head and a non-empty body.*

Observe that a filtering rule can be either strict or defeasible and, as we will explain below, the effect in the filtering process will be different. Note also that a filtering rule cannot be a single literal (*i.e.*, a fact). Below we will explain how to use filtering rules in order to select desires, but first we will introduce an example to provide some motivation.

Example 3 *Considering the set* D *introduced in Example 2, the following filtering rules can be defined for selecting desires:*

$$\Pi_F = \left\{ \begin{array}{l} \sim carry \leftarrow \sim hasBall \\ \sim shoot \leftarrow \sim hasBall \end{array} \right\}$$

$$\Delta_F = \left\{ \begin{array}{l} shoot \prec theirGoalieAway \\ carry \prec noOneAhead \\ \sim shoot \prec farFromGoal \\ \sim carry \prec shoot \end{array} \right\}$$

Consider a particular situation in which the agent does not have the ball (*i.e.*, $\sim hasBall$ holds as a belief), then from the agent's beliefs and the filtering rules (Π_F,Δ_F) of Example 3 there are warrants for $\sim carry$ and $\sim shoot$. In this situation the agent should not consider selecting the desires $carry$ and $shoot$ because there are warranted reasons against them.

Suppose now another situation in which the agent has the ball and the opponent goalie is away from its position but the agent is far from the goal (*i.e.*, $\{theirGoalieAway$, $farFromGoal\} \subset$ B). Then, from the agent's beliefs and the filtering rules (Π_F,Δ_F) of Example 3, there are arguments for both $shoot$ and $\sim shoot$. Since these two arguments defeat each other, a blocking situation occurs and the answer for each one is UNDECIDED. In contrast with the previous situation, here there are no strong reasons against selecting the desire $shoot$, and as we will show below, in this formalism an undecided desire will be eligible.

In Definition 3 below, we will introduce a mechanism for filtering the set of desires D in order to obtain a subset of D containing only those desires achievable in the *current* situation. In order to do that, beliefs and filtering rules should be used in combination. Hence, we need to explain how two defeasible logic programs can be properly combined.

Combining beliefs with filtering rules

In this formalism, agents will have a defeasible logic program (*de.l.p.*) (Π_B,Δ_B) containing rules and facts for deriving beliefs, and a *de.l.p.* (Π_F,Δ_F) with filtering rules for selecting desires. Observe that the union of two *de.l.p.* might not be a *de.l.p.*, because the union of two sets of consistent strict rules could be contradictory (*e.g.*, Π_F=$\{(a \leftarrow b)$, $(\sim a \leftarrow c)\}$ and Π_B=$\{b, c\}$). Therefore, in order to guarantee that $\Pi_B \cup \Pi_F$ is not contradictory, a merge revision operator is needed. Therefore, instead of simply having ($\Pi_B \cup \Pi_F$, $\Delta_B \cup \Delta_F$) we will impose to have ($\Pi_B \circ \Pi_F$, $\Delta_B \cup \Delta_F \cup \Delta_X$), where "$\circ$" is a merge revision operator (Fuhrmann 1997). The mechanism of this operator (Falappa, Kern-Isberner, & Simari 2002) "is to add Π_B to Π_F and then eliminate from the result all possible inconsistency by means of an incision function that makes a *cut* over each minimally inconsistent subset of $\Pi_B \cup \Pi_F$."

Observe that in particular in our approach the set Π_F will contain only strict rules representing filtering rules, whereas the set Π_B will contain beliefs representing the agent perception of the environment. Therefore, the elements of Π_B can not be ignored or deleted. Hence, the conflicting set X of $\Pi_B \circ \Pi_F$ will be defined by eliminating all the conflicting strict rules from Π_B and Π_F. Thus, if literals L and $\sim L$ can be derived from $\Pi_B \cup \Pi_F$, those strict rules that have L or $\sim L$ in their heads will be in X.

The merge revision operator will remove the inconsistency from $\Pi_B \cup \Pi_F$ on behalf of some criterion. This could be achieved, for example, by deleting all the rules in conflict, but the part of the knowledge represented by these rules will be lost. Another option is to convert the status of the (strict) rules involved in the conflict, turning them into defeasible rules (Falappa, Kern-Isberner, & Simari 2002). This criterion intends to keep the encoded information that would be, otherwise, lost.

Therefore, in our case, the union of two *de.l.p.* like (Π_B,Δ_B) and (Π_F,Δ_F) will be a program (Π,Δ), where Π = $\Pi_B \circ \Pi_F$ and Δ = $\Delta_B \cup \Delta_F \cup \Delta_X$. The set Π is obtained using a merge revision operator $\circ$ that eliminates X, and the set Δ_X is a set of defeasible rules obtained transforming each strict rule $r \in X$ in a defeasible rule.

Example 4 *Here we show how the merge revision operator works. Consider having the following sets:*

Π_F=$\{(a \leftarrow b)$, $(\sim a \leftarrow c)\}$

Π_B=$\{b, c\}$

Note that in this case $\Pi_F \cup \Pi_B$ is a contradictory set. Therefore, it cannot be part of a valid de.l.p. *As stated above, we will apply the merge revision operator instead of performing the union of both sets. Hence, we have:*

$\Pi_B \circ \Pi_F = \{b, c\}$

Δ_X=$\{(a \prec b)$, $(\sim a \prec c)\}$

Thus, after the application of the merge operator, no knowledge is lost (although it is certainly weakened), because it is now encoded under the form of defeasible rules.

Selecting desires

The next definition introduces a mechanism for filtering the set of desires D in order to obtain a subset containing only those desires achievable in the *current* situation.

Definition 3 (Current desires) *Let $K = (\Pi_B \circ \Pi_F, \Delta_B \cup \Delta_F \cup \Delta_X)$ be a defeasible logic program where (Π_B,Δ_B) contains rules and facts for deriving beliefs, and (Π_F,Δ_F) contains filtering rules. Given a set* D *of desires, the set* D^c *of* Current Desires *will be defined as:*

$$D^c = \{\delta \in D \mid \textit{there is no warrant for } \overline{\delta} \textit{ from } K\}$$

Thus, the set D^c will be a subset of the set D and, at any given moment, it will contain those desires that have a chance of being achieved. Observe that a literal L will not belong to D^c when its complement $\overline{L}$ is warranted.

Example 5 *Let's consider a subset of the set of desires from example 2:*

$$D = \left\{ \begin{array}{ll} shoot & \sim shoot \\ carry & \sim carry \\ goto(Place) & \sim goto(Place) \end{array} \right\}$$

Taking the filtering rules from example 3:

$$\Pi_F = \left\{ \begin{array}{l} \sim carry \leftarrow \sim hasBall \\ \sim shoot \leftarrow \sim hasBall \end{array} \right\}$$

$$\Delta_F = \left\{ \begin{array}{l} shoot \prec theirGoalieAway \\ carry \prec noOneAhead \\ \sim shoot \prec farFromGoal \\ \sim carry \prec shoot \end{array} \right\}$$

Then, if we consider the following perceived beliefs:

$$\Pi_{\mathsf{B}} = \Phi = \left\{ \begin{array}{l} theirGoalieAway \\ noOneAhead \end{array} \right\}$$

We will have these current desires:

$$\mathsf{D}^c = \left\{ \begin{array}{l} shoot \\ carry \\ \sim carry \\ goto(Place) \\ \sim goto(Place) \end{array} \right\}$$

Using the filtering rules from (Π_F,Δ_F), the argument of Figure 2 for $shoot$ can be built. This argument has no defeaters, therefore, its conclusion $shoot$ is warranted (*i.e.*, the DeLP answer is YES). Hence, $shoot$ will belong to the set D^c of current desires. An argument for desire $carry$ also exists, but it is blocked by a counter-argument holding $\sim carry$, as shown in Figure 3; then, both $carry$ and $\sim carry$ DeLP answers are UNDECIDED, and they are included into D^c. Finally, considering the desires $goto(Place)$ and $\sim goto(Place)$, we can see that there are no filtering rules regarding to them, so their DeLP answers are UNKNOWN, and both will belong to D^c.

Figure 2: *Undefeated argument for* $shoot$.

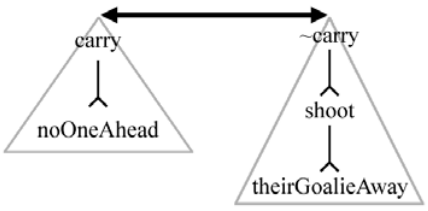

Figure 3: *Blocking situation for* $carry$ *and* $\sim carry$.

You may notice that if neither Q nor $\overline{Q}$ has a warrant built from ($\Pi_{\mathsf{B}} \circ \Pi_F$, $\Delta_{\mathsf{B}} \cup \Delta_F \cup \Delta_X$), then both will be included into the set D^c. Therefore, the agent will have to consider these two options (among the rest of the current desires) in order to choose an intention, although they are in contradiction.

In this work we will require having B and D as two separate sets. If a literal is allowed to belong to both sets, then, when joining the (Π_{B},Δ_{B}) and (Π_F,Δ_F) *de.l.p.* programs, an undesirable mix of concepts would arise. Note that this is not a strong restriction, because the fact that a literal could be both a belief and a desire brings about well-known representational issues.

Remark 1 *We will assume that no literal* $L \in \mathsf{D}$ *belongs to* B. *That is, a desire cannot be perceived or derived as a belief.*

Example 6 *Here we will show why it is important to take Remark 1 into consideration, keeping literals from* B *and* D *apart. Consider the following sets of beliefs and desires:*

$$(\Pi_{\mathsf{B}},\Delta_{\mathsf{B}}) = (\{x, y\},\{a \prec y\})$$
$$\mathsf{D} = \{a, \sim a\}$$

And we will consider these filtering rules:

$$(\Pi_F,\Delta_F) = (\{\},\{\sim a \prec x\})$$

Here, $\Pi_{\mathsf{B}} \circ \Pi_F = \{x, y\}$*; and* $\Delta_{\mathsf{B}} \cup \Delta_F \cup \Delta_X = \{a \prec y, \sim a \prec x\}$. *We have that '*$\sim a$*' has an argument based on* Δ_F*, and '*a*' has an argument built from* (Π_{B},Δ_{B}). *By Definition 3, both literals will belong to* D^c. *However, if we let this happen, we will be letting the beliefs rules decide which desires are going to be in* D^c. *Although the argumentation process involved in the selection of the current desires potentially requires some dialectical analysis to be performed upon the rules defined in* (Π_{B},Δ_{B})*, the elements of* D^c *should be determined only by the filtering rules in* (Π_F,Δ_F).

A simple way of satisfying the restriction imposed by Remark 1 could be to distinguish literals in D with a particular predicate like "$desire$". For example: $desire(shoot)$, $\sim desire(shoot)$, $desire(pass(Mate))$, *etc.* Thus, assuming that no belief is represented with the predicate name "$desire$", then literals representing beliefs will be, by construction, different from the ones representing desires. Although this alternative is supported by this formalism, in the examples given in this paper we will use the convention of having different names for desires and beliefs.

The set D^c can be also defined in terms of DeLP answers. As stated in the last section about DeLP, given a literal Q there are four possible answers for the query Q: YES, NO, UNDECIDED, and UNKNOWN. Thus, given $Q \in \mathsf{D}$, Q will be in D^c if, from ($\Pi_{\mathsf{B}} \circ \Pi_F$, $\Delta_{\mathsf{B}} \cup \Delta_F \cup \Delta_X$), the answer for Q is YES, UNDECIDED or UNKNOWN.

Next, we introduce several propositions and properties that show how different characteristics of both D and (Π_F,Δ_F) determine the contents of D^c.

Proposition 3 *Given a literal* $Q \in \mathsf{D}$*, if there is no filtering rule in* (Π_F,Δ_F) *with head* Q *nor its complement* $\overline{Q}$*, then* $\{ Q, \overline{Q} \} \subseteq \mathsf{D}^c$.

Proof: If $Q \in \mathsf{D}$*, then* $\overline{Q} \notin \mathsf{B}$ *(remark 1), and since* $\overline{Q}$ *is not in the head of any filtering rule, then it is not possible to obtain a warrant for* $\overline{Q}$*. Therefore, by Definition 3,* Q *will be included into the set* D^c *of current desires.*

Consider the sets Π_F, Δ_F, Π_{B} and D from Example 5; we have that $goto(Place)$ and $\sim goto(Place)$ belong to D, but there are no filtering rules in (Π_F,Δ_F) with head $goto(Place)$ nor $\sim goto(Place)$. Note that both literals are in D^c, which corresponds with Proposition 3.

Observe that the proof of Proposition 3 can also be expressed in terms of DeLP answers. Since there are no rules with head Q nor its complement, and $\overline{Q} \notin \mathsf{B}$, then Q is not in the language of ($\Pi_{\mathsf{B}} \circ \Pi_F$, $\Delta_{\mathsf{B}} \cup \Delta_F \cup \Delta_X$). Therefore, from ($\Pi_{\mathsf{B}} \circ \Pi_F$, $\Delta_{\mathsf{B}} \cup \Delta_F \cup \Delta_X$) the answer for Q will be UNKNOWN (the same holds for $\overline{Q}$). Because these answers

are different from NO, Q and $\overline{Q}$ will be included into the set D^c of current desires.

Proposition 4 *Given a literal $Q \in \mathsf{D}$, if Q is warranted from $K = (\Pi_\mathsf{B} \circ \Pi_F, \Delta_\mathsf{B} \cup \Delta_F \cup \Delta_X)$, then $Q \in \mathsf{D}^c$ and $\overline{Q} \notin \mathsf{D}^c$.*

Proof: Since Q has a warrant built from K, then there is no warrant for $\overline{Q}$, therefore, $Q \in \mathsf{D}^c$. To prove that $\overline{Q} \notin \mathsf{D}^c$, we have to show that there is a warrant for the complement of $\overline{Q}$ (i.e., Q) which is true by hypothesis.

Considering the case shown in Example 5, see that desire $shoot$ has a warrant built from $(\Pi_\mathsf{B} \circ \Pi_F, \Delta_\mathsf{B} \cup \Delta_F \cup \Delta_X)$, and $shoot$ belongs to D^c, but $\sim shoot$ does not, which coincides with the assertion of Proposition 4.

The next proposition shows that the set D^c of current desires will not be empty if the set D of desires is not empty. This does not depend on the set of filtering rules nor the set of beliefs.

Proposition 5 *For any $K = (\Pi_\mathsf{B} \circ \Pi_F, \Delta_\mathsf{B} \cup \Delta_F \cup \Delta_X)$, D will not be empty if, and only if, D^c is not empty.*

Proof:

($\Rightarrow$) Let L and $\overline{L}$ be two elements of D, then one of the three following cases holds:
(a) L is warranted from K, therefore, $L \in \mathsf{D}^c$.
(b) $\overline{L}$ is warranted from K, therefore, $\overline{L} \in \mathsf{D}^c$
(c) Neither L nor $\overline{L}$ is warranted from K, therefore, both L and $\overline{L}$ will belong to D^c.
Hence, in any case, D^c will not be empty.
($\Leftarrow$) Trivial from Definition 3. That is, if D^c is not empty, then D cannot be empty.

Given an agent with $K = (\Pi_\mathsf{B} \circ \Pi_F, \Delta_\mathsf{B} \cup \Delta_F \cup \Delta_X)$, and a set of desires D, the following properties hold:

1. *For every literal $L \in \mathsf{D}$, it cannot be the case that neither L nor $\overline{L}$ do not belong to D^c.*
   Suppose that L and $\overline{L}$ are not in D^c, then, the complement of each (*i.e.*, $\overline{L}$ and L) must be warranted from K, which is not allowed in DeLP. Observe that, if both literals are warranted, then both literals have to belong to D^c, which contradicts the initial supposition.
2. *If there are no filtering rules then $\mathsf{D}^c = \mathsf{D}$.*
   This is a particular case of Proposition 3. If there are no filtering rules (*i.e.*, $\Pi_F = \emptyset$ and $\Delta_F = \emptyset$) then no element of D will have its complement warranted, so every element of D will be in D^c.
3. *If there are no desires warranted from K, then $\mathsf{D}^c = \mathsf{D}$.*
   If no element of D is warranted from K, then no element of D will have its complement warranted from K ($L \in \mathsf{D}$ implies that $\overline{L} \in \mathsf{D}$). Thus, every element of D will be in D^c.
4. *If there is, at least, one element of D that is warranted from K, then D^c will be a proper subset of D.*
   If L belongs to D and is warranted from K, then $\overline{L} \notin \mathsf{D}^c$.

Application to Robotic Soccer

Robotic soccer has proven to be a system complex enough to test many of the features of any reasoning system. The robots are controlled by software agents, each of which has a set of high-level actions to perform, such as kicking the ball with a given strength or moving in a given direction. At every moment, an agent has to choose which action to do next. That choice can be made using a reasoning system, in this case, a defeasible argumentation system.

In this Section we will show how a player makes a decision based on the model proposed in this paper. The scenario has three players belonging to one team ('self', 't1' and 't2' in Figure 4) and three players from the opposite team ('o1', 'o2' and 'o3' in Figure 4). We will analyze the reasoning performed by the player named 'self'. Regarding knowledge representation, the belief predicates will be written according to the Close World Assumption, and everything that cannot be proved will be assumed false.

Let's suppose that we have a soccer-playing agent, with a desires set including the options $shoot$, $pass$ and $carry$, being in the situation depicted in Figure 4. The agent will perceive the positions of the ball and all the other players, and will reason about what to do next. The agent will have different rules, like $pass(self, t1) \prec marked(self)$, that will be a reason for passing the ball to $t1$.

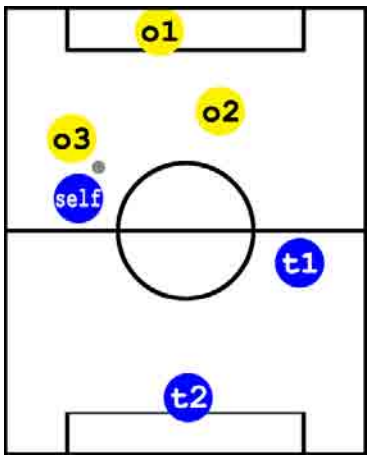

Figure 4: *'self' decides to pass the ball to 't1'*

In this situation, the agent has the following perceptions:

$$\Phi = \left\{ \begin{array}{l} marked(self), \\ oppositeField(self), \\ noOneAhead(t1), \\ hasBall(self), \\ betterPosition(self, t1) \end{array} \right\}$$

The knowledge is represented through a *de.l.p.*, defining reasons for and against every element belonging to the set D via strict and defeasible rules. In this example, the sets Δ_F and Π_F are the following:

$\Pi_F = \{$
$(\sim shoot(P) \leftarrow \sim hasBall(P))$,
$(\sim pass(Src, \_) \leftarrow \sim hasBall(Src))$,

$(\sim carry(P) \leftarrow \sim hasBall(P))$ }

Δ_F= {
$(shoot(P) \prec opposite Field(P), noOneAhead(P))$,
$(shoot(P) \prec opposite Field(P), not\ marked(P))$,
$(shoot(\_) \prec goalieAway(opposite))$,

$(\sim shoot(P) \prec pass(P, \_))$,
$(\sim shoot(P) \prec carry(P))$,

$(pass(Src, \_) \prec marked(Src))$,
$(pass(Src, Tgt) \prec betterPosition(Tgt, Src))$,

$(\sim pass(Src, Tgt) \prec playerBetween(Src, Tgt))$,
$(\sim pass(\_, Tgt) \prec marked(Tgt))$,
$(\sim pass(Src, \_) \prec shoot(Src))$,
$(\sim pass(Src, \_) \prec carry(Src))$,

$(carry(P) \prec noOneAhead(P))$,

$(\sim carry(P) \prec shoot(P))$,
$(\sim carry(P) \prec pass(P, \_))$ }

At the particular moment of Figure 4, the player has the following Beliefs set:

$$\mathsf{B} = \left\{ \begin{array}{l} marked(self), \\ oppositeField(self), \\ noOneAhead(t1), \\ hasBall(self), \\ betterPosition(self, t1), \\ teammate(t1), \\ myRole(forward) \end{array} \right\}$$

Observe that some of these beliefs are perceived (*e.g.*, $marked(self)$), while other are persistent (*e.g.*, $teammate(t1)$).

Once built the internal representation of the world, the agent performs DeLP queries over the elements of its set D, gathering their corresponding answers:

- *Shooting on goal:* If you consider program $\mathcal{P}$, along with the set B of Beliefs, it is clear that no argument can be built for the desire $shoot(self)$, because every rule with head $shoot(self)$ has at least one literal in its body that does not hold. As you can see in Figure 4, there are no significant common-sense reasons to shoot on goal.

  On the other hand, an argument for $\sim shoot(self)$ can be constructed from $\mathcal{P}$: the one built upon the argument of being able to perform a pass to a teammate. That counter-argument is undefeated: there are no reasons against it. Figure 4 illustrates this situation, and shows that passing the ball to player 't1' seems to be the better choice to make at that moment.

  The answer given from the interpreter is NO and the agent will no longer consider shooting on goal.

- *Passing the ball:* From program $\mathcal{P}$, one argument can be stated for $pass(self, t1)$, based on the reason that player 'self' is marked. None of the rules that could build an argument against it holds, so this one is undefeated. As explained above, from the situation we are considering, it is reasonable to think that this may be a good choice.

  The answer given from the interpreter is YES and the agent will have a strong reason to pass the ball to 't1'.

- *Carrying the ball:* there are no rules from which an argument for carrying the ball could be built. Program $\mathcal{P}$ and Figure 4 coincide in this matter: there seems to be no reason for carrying the ball. On the other hand, the aforementioned argument for passing the ball is an undefeated reason for not doing it.

  The interpreter answers NO and the agent will no longer consider this desire.

Once gathered all the DeLP answers, the set D^c of current desires can be built:

$$\mathsf{D}^c = \{\ pass(self, t1)\ \}$$

This means that, in this situation, the player has a clear choice: the selected intention must be *perform a pass to 't1'*, because it is the only current desire and it is warranted.

Related work

In a very recent paper (Rahwan & Amgoud 2006), Rahwan and Amgoud have proposed an argumentation-based approach for practical reasoning that extends (Amgoud 2003) and (Amgoud & Cayrol 2002), introducing three different instantiations of Dung's abstract argumentation framework in order to reason about beliefs, desires and plans, respectively. This work is, in our concern, the one that is most related to our approach and, as we will show next, it have many points in common with our work. Both approaches use a defeasible argumentation formalism for reasoning about beliefs and desires (in their work, they also reason about plans, but this is out of the scope of our presentation). Like us, they separate in the language those rules for reasoning about belief from those rules for reasoning about desires; and, in both approaches, it is possible to represent contradictory information about beliefs and desires. Since both approaches use a defeasible argumentation formalism, they construct arguments supporting competing desires, and these arguments are compared and evaluated in order to decide which one prevails. Their notion of *desire rule* is very similar to our notion of *filtering rule*.

Although both approaches have many things in common, they also differ in many points. In their case, two different argumentation frameworks are needed in order to reason about desires: one framework for beliefs rules and other framework for desires rules. The last one depends directly on the first one, and since there are two kinds of arguments, a policy for comparing mixed arguments is given. In our case, only one argumentation formalism is used for reasoning with both types of rules. In their object language, beliefs and desires include a certainty factor for every formula, and no explicitly mention of perceived information is given. In our case, uncertainty is represented by defeasible rules (see (García & Simari 2004)) and perceived beliefs are explicitly treated by the model. Besides, although both approaches use defeasible argumentation, the argumentative formalism used in their approach differs from ours. In their case, the comparison of arguments relies on the certainty

factor given to each formula, and they do not distinguish between proper and blocking defeaters.

The use of defeasible argumentation for reasoning in BDI architectures is not new and it was originally mention in the seminal paper (Bratman, Israel, & Pollack 1991) and in other more recent works like (Parsons, Sierra, & Jennings 1998). Other related work includes (Thomason 2000) and (Broersen *et al.* 2001), where a formalism for reasoning about beliefs and desires is given. However, these last formalisms differ from our because they do not use argumentation.

Conclusions and future work

In this paper we have shown how a deliberative agent can represent its perception and beliefs using a defeasible logic program. The information that the agent perceives directly from its environment is represented with a subset of *perceived beliefs* that is updated dynamically, and a subset Π formed with strict rules and facts represent other static knowledge of the agent. Defeasible argumentation is used in order to warrant agents (derived) beliefs.

Strict and defeasible filtering rules have been introduced to represent knowledge for selecting desires and a defeasible argumentation can be used for selecting a proper desire that fits in the particular situation the agent is involved.

With this formalism, agents can reason about its desires in order to select the appropriate ones. However, following the spirit of the BDI model, once appropriate desires are detected, the agent should select (and commit to) a specific intention (goal), and then select appropriate actions to fulfill that intention. We are currently working in extending this approach to consider the representation of agent's intentions and reasoning about them.

References

Amgoud, L., and Cayrol, C. 2002. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence* 34(1-3):197–215.

Amgoud, L. 2003. A formal framework for handling conflicting desires. In *Proceedings of the 7th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU'2003*, 552–563.

Bratman, M. E.; Israel, D.; and Pollack, M. 1991. Plans and resource-bounded practical reasoning. In Cummins, R., and Pollock, J. L., eds., *Philosophy and AI: Essays at the Interface*. Cambridge, Massachusetts: The MIT Press. 1–22.

Broersen, J.; Dastani, M.; Hulstijn, J.; Huang, Z.; and van der Torre, L. 2001. The boid architecture: conflicts between beliefs, obligations, intentions and desires. In *Proceedings of Fifth International Conference on Autonomous Agents (Agents2001)*. Montreal, Canada: ACM Press. 9–16.

Chesñevar, C. I.; Maguitman, A. G.; and Loui, R. P. 2000. Logical Models of Argument. *ACM Computing Surveys* 32(4):337–383.

Falappa, M. A.; Kern-Isberner, G.; and Simari, G. R. 2002. Belief revision, explanations and defeasible reasoning. *Artificial Intelligence Journal* 141:1–28.

Fuhrmann, A. 1997. *An Essay on Contraction*. Studies in Logic, Language and Information, CSLI Publications, Stanford, California.

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

Lifschitz, V. 1996. Foundations of logic programming. In Brewka, G., ed., *Principles of Knowledge Representation*. CSLI. 69–127.

Parsons, S.; Sierra, C.; and Jennings, N. 1998. Agents that reason and negotiate by arguing. *Journal of Logic and Computation* 8(3):261–292.

Prakken, H., and Vreeswijk, G. 2002. Logical systems for defeasible argumentation. In D.Gabbay., ed., *Handbook of Philosophical Logic, 2nd ed.* Kluwer Academic Pub.

Rahwan, I., and Amgoud, L. 2006. An argumentation-based approach for practical reasoning. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*.

Rao, A. S., and Georgeff, M. P. 1995. BDI-agents: from theory to practice. In *Proceedings of the First International Conference on Multiagent Systems*.

Rao, A. S. 1996. AgentSpeak(L): BDI agents speak out in a logical computable language. In van Hoe, R., ed., *Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World*.

Simari, G. R., and Loui, R. P. 1992. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence* 53(2–3):125–157.

Thomason, R. 2000. Desires and defaults: A framework for planning with inferred goals. In *Proceedings of the seventh international Confenrence on Principle of Knowledge Representation and Reasoning (KR'00)*, 702–713.