Reasoning from Desires to Intentions: A Dialectical Framework

Nicolás D. Rotstein and **Alejandro J. García** and **Guillermo R. Simari**
National Council of Scientific and Technical Research (CONICET)
Artificial Intelligence Research and Development Laboratory
Department of Computer Science and Engineering
Universidad Nacional del Sur, Bahía Blanca, Argentina
e-mail: {ndr,ajg,grs}@cs.uns.edu.ar

Abstract

Here, we define a framework where defeasible argumentation is used for reasoning about beliefs, desires and intentions. A dialectical filtering process is introduced to obtain a subset of the agent's desires containing only those that are achievable in the current situation. Different agents types can be defined in the framework affecting the way in which current desires are obtained.

The agent is provided with a set of intention rules that specifies under what conditions an intention could be achieved. When more than one intention is present, a policy will be used to choose among them. Thus, intention policies provide the agent with a mechanism for deciding which intention is selected in the current situation. Several application examples will be given.

Introduction

In this work, we introduce a framework where defeasible argumentation is used for warranting agent's beliefs, filtering desires, and selecting proper intentions according to a given policy. In our framework, different types of agents can be defined and this decision will affect the way in which desires are filtered.

Autonomous agents based on mental attitudes had gathered special attention in the last years, specially those that follow architectures based on BDI. There are several approaches built upon BDI, some of them introducing new components, like the BOID architecture (Broersen *et al.* 2001). Regarding the underlying logic of the sets of *Beliefs*, *Desires* and *Intentions*, some approaches are based on theories such as Default Logic (or Normal Default Logic) (Broersen *et al.* 2001; Thomason 2000), whereas others introduce a formal framework that combines BDI with an argumentation formalism (Amgoud 2003; Hulstijn & van der Torre 2004; Parsons, Sierra, & Jennings 1998; Rahwan & Amgoud 2006).

In (Rotstein & García 2006) a proposal for using defeasible argumentation to reason about agent's beliefs and desires was introduced. There, a mechanism is described to filter agent's desires to obtain a set of current desires, *i.e.*, those that are achievable in the current situation, but in that work agent's intentions were not considered. Here, we extend that approach by adding agent's intentions, and propose a new, more general filtering process that involves the introduction of the notion of *agent type*. Thus, the definition of the filtering process is more flexible, and the way in which desires are filtered will be determined by the agent type. We introduce an agent architecture (see Fig. 1) where intentions are considered and the relationship between the different components (B, D, and I) is made explicit. To select an intention to achieve, intention rules and a selection policy will be defined.

The idea of using defeasible argumentation in the reasoning process of BDI agents is not new (Bratman, Israel, & Pollack 1991), and there exist previous approaches that relate BDI with abstract argumentation frameworks (Amgoud 2003; Parsons, Sierra, & Jennings 1998; Rahwan & Amgoud 2006). The contribution of our approach is to introduce a BDI architecture that uses a concrete framework based on a working defeasible argumentation system: Defeasible Logic Programming (DeLP). As will be shown next (see Fig. 1), argumentation will be used for reasoning about beliefs, for filtering desires and for selecting intentions following a given policy. We provide meaningful examples from a robotic soccer domain to show how an agent can be implemented using this framework.

An outline of the proposed framework appears in Fig. 1. Briefly, the main input is the *perception* from the environment, which is part of the set of *belief rules* (Π_{B},Δ_{B}) that, through an *argumentation process*, leads to the set B of warranted beliefs. We will describe how these belief rules are used with a set of *filtering rules* (Π_F,Δ_F). This new set, along with a set D of possible desires and the specification of a *filtering function* are the input to a *dialectical filtering process*, whose output is the set D^c of the agent's *current desires*. The final stage of the *agent behavior loop* involves the usage of a set of *intention rules*, embedded in an *intention policy* that will determine the preferred rule. The current desire in the head of this rule will be the *selected intention*.

As shown in Fig. 1, there are three main processes. They use defeasible argumentation based on Defeasible Logic Programming (DeLP). Next, we give a brief summary of DeLP (for more details see (García & Simari 2004)). In DeLP, knowledge is represented using facts, strict rules, and defeasible rules:

Partially supported by CONICET, ANPCyT, and UNSur.

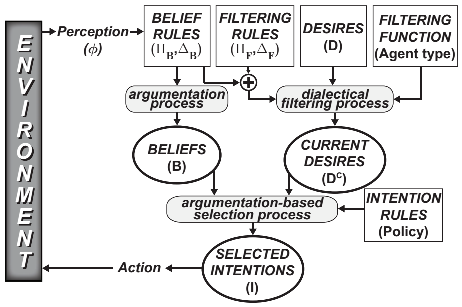

Figure 1: DeLP-based BDI architecture

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

Rules are distinguished by the type of arrows, and a defeasible rule "$Head \prec Body$" expresses that "*reasons to believe in the antecedent $Body$ give reasons to believe in the consequent $Head$*" representing tentative information that may be used if nothing could be posed against it.

A Defeasible Logic Program (*de.l.p.*) $\mathcal{P}$ 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. Strict and defeasible rules are ground, however, following the usual convention (Lifschitz 1996), some examples will use "schematic rules" with variables.

Strong negation could appear in the head of program rules, and can be used to represent contradictory knowledge. From a program (Π, Δ) contradictory literals could be derived, however, the set Π (used to represent non-defeasible information) must be non-contradictory, *i.e.*, no pair of contradictory literals can be derived from Π. Given a literal L, $\overline{L}$ represents the complement with respect to strong negation. If contradictory literals are derived from (Π, Δ), a dialectical process is used for deciding which literal prevails. 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 $\mathcal{A} \cup \Pi$. A literal L is *warranted* from (Π, Δ) if there exists a non-defeated argument $\mathcal{A}$ supporting L. 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 $\langle \mathcal{A}, L \rangle$. A defeater $\mathcal{A}_1$ for an argument $\mathcal{A}_2$ can be proper ($\mathcal{A}_1$ *stronger than* $\mathcal{A}_2$) or *blocking* (same strength). In the examples that follow we assume generalized specificity as the comparison criterion, however, as explained in (García & Simari 2004) the criterion could be easily changed.

Since defeaters 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, a query Q could have four possible answers: 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 signature of the program.

Warranting beliefs

Following (Rotstein & García 2006), agent's beliefs correspond to the semantics[1] of a *defeasible logic program* $\mathcal{P}_{\mathsf{B}} = (\Pi_{\mathsf{B}},\Delta_{\mathsf{B}})$. In Π_{B} two disjoint subsets will be distinguished: Φ of *perceived beliefs* that will be updated dynamically (see Fig. 1), and Σ of strict rules and facts that will represent static knowledge, $\Pi_{\mathsf{B}} = \Phi \cup \Sigma$. Besides the perceived beliefs, the agent may use strict and defeasible rules from $\mathcal{P}_{\mathsf{B}}$ to obtain a *warrant* for its *derived beliefs* (see Definition 1).

Since Π_{B} has to be non-contradictory, we assume that perception is correct in the sense that it will not give a pair of contradictory literals. We will also require that no perceived literal in Φ can be derived directly from Σ. Thus, if Σ 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 $(\Pi_{\mathsf{B}},\Delta_{\mathsf{B}})$.

Definition 1 (Belief types) *A **Perceived belief** is a fact in Φ 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 $\Pi_{\mathsf{B}} = \Phi \cup \Sigma$ (i.e., no defeasible rules are used for its derivation). A **Defeasible belief** is a warranted literal L supported by an non-empty argument $\mathcal{A}$ (i.e., it uses at least one defeasible rule). 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 $\mathsf{B} = \Phi \cup \mathsf{B}_s \cup \mathsf{B}_d$.*

Example 1 *Consider a robotic-soccer agent Ag that has the following program $(\Pi_{\mathsf{B}},\Delta_{\mathsf{B}})$, where Π_{B} was divided distinguishing the set Φ=$\{hasBall(t1),\ marked(t1)\}$ of perceived facts representing "player $t1$ has the ball", and "teammate $t1$ is marked", the set Σ of non-perceived information, and the set Δ_{B} of defeasible knowledge:*

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

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

Observe that Ag can infer the strict belief: $\sim mate(o1)$.

The argument built from $(\Pi_{\mathsf{B}},\Delta_{\mathsf{B}})$ for $receive(self)$: $\{receive(self) \prec hasBall(t1), mate(t1)\}$, has no defeaters, and therefore, there is a warrant for one defeasible belief: $receive(self)$ (Ag may receive a pass).

The sets Φ, B_s and B_d are disjoint sets. It can be shown that the set B of beliefs of an agent is a non-contradictory set of warranted literals. Although perceived beliefs are facts in Π_{B}, there could be other facts in Π_{B} which are not perceived,

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

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 cannot change with perception, like $myRole(defender)$, or $mate(t1)$.

We assume a perception function that provides the agent with information about its environment. This function will be invoked by the agent to update its perceived beliefs set Φ. When this happens the new information overrides the old one following some criterion. Updating a set of literals is a well-known problem and many proposals exist in the literature (Falappa, Kern-Isberner, & Simari 2002; Fuhrmann 1997).

Example 2 *In the context of Ex. 1, with the perception that the agent is now marked, the set* Φ *becomes* $\{hasBall(t1), marked(t1), marked(self)\}$. *Now the argument for* $receive(self)$ *has a "blocking defeater", which means that the DeLP answer for both* $receive(self)$ *and* $\sim receive(self)$ *will be* UNDECIDED.

Consider another situation, where $\Phi = \{hasBall(o1)\}$. *Here, the answer for* $receive(self)$ *is* NO, *since there is a warrant for* $\sim receive(self)$ *supported by the non-defeated argument* $\{\sim receive(self) \prec hasBall(o1), \sim mate(o1)\}$.

Filtering Desires

Agents desires will be represented by a given set D of literals that will contain a literal representing each desire the agent might want to achieve. Clearly, D may be contradictory, that is, both a literal L and its complement $\overline{L}$ might belong to D. We will assume that beliefs and desires are represented with separate names, *i.e.*, $\mathsf{D} \cap \mathsf{B} = \emptyset$. Hence, a desire cannot be perceived or derived as a belief.

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 impossible to be carried out. For example, if the agent does not have the ball and the ball is in a place p, then, the desire $shoot$ could not be effected, whereas $goto(p)$ is a plausible option. Therefore, agents should reason about their desires to select the ones that could be actually realized. 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 (see Figure 1).

In (Rotstein & García 2006) a reasoning formalism was introduced for selecting from D those desires that are suitable to be brought about. To perform this selection, the agent uses its beliefs (representing the current situation) and a defeasible logic program (Π_F,Δ_F) composed by *filtering rules*. The filtering rules represent reasons for and against adopting desires. In other words, filtering rules eliminate those desires that cannot be effected in the situation at hand. Once the set of achievable desires is obtained, the agent can adopt one of them as an intention.

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 will be explained below, that will influence the filtering process. 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 *A robotic-soccer agent could have the following sets of desires and filtering rules:*

$$\mathsf{D} = \left\{ \begin{array}{l} shoot \\ carry \\ pass \\ move \end{array} \right\} \Pi_F = \left\{ \begin{array}{l} \sim carry \leftarrow \sim ball \\ \sim shoot \leftarrow \sim ball \\ \sim pass \leftarrow \sim ball \end{array} \right\}$$

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

Consider a particular situation in which an agent does not have the ball (*i.e.*, $\sim ball \in \Phi$). If the agent has $\Delta_{\mathsf{B}} = \emptyset$, $\Pi_{\mathsf{B}} = \Phi$ and the filtering rules (Π_F,Δ_F) from Ex. 3, then, there are warrants for $\sim carry$, $\sim pass$ and $\sim shoot$ from this information. Hence, in this particular situation, the agent should not consider selecting the desires $carry$, $pass$, and $shoot$, because there are justified reasons against them. Observe that these reasons are not defeasible.

Suppose now a new set of perceived beliefs: $\mathsf{B} = \Phi = \{ball, theirGoalieAway, farFromGoal\}$, that is, 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. Then, from the agent's beliefs and the filtering rules (Π_F,Δ_F) of Ex. 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 both literals is UNDECIDED. In our approach (as will be explained later) an undecided desire could be eligible.

In this formalism, beliefs and filtering rules should be used in combination. Hence, we need to explain how two defeasible logic programs can be properly combined. Agents will have a *de.l.p.* $(\Pi_{\mathsf{B}},\Delta_{\mathsf{B}})$ containing rules and facts for deriving beliefs, and a *de.l.p.* (Π_F,Δ_F) with filtering rules for selecting desires. We need to combine these two *de.l.p.*, but the union of them might not be a *de.l.p.*, because the union of the sets of strict rules could be contradictory. To overcome this issue, we use a merge revision operator "$\circ$" (Fuhrmann 1997). Hence, in our case, the join of $(\Pi_{\mathsf{B}},\Delta_{\mathsf{B}})$ and (Π_F,Δ_F) will be a program (Π,Δ), where $\Pi = \Pi_{\mathsf{B}} \circ \Pi_F$ and $\Delta = \Delta_{\mathsf{B}} \cup \Delta_F \cup \Delta_X$. A set X is introduced, containing those strict rules r_i that derive complementary literals. This set is eliminated when merging Π_{B} and Π_F, then every r_i is transformed into a defeasible rule, and the set Δ_X is generated, carrying the resulting defeasible rules (see (Rotstein & García 2006) for more details).

Definition 3 (Agent's Knowledge Base) *Let* $(\Pi_{\mathsf{B}},\Delta_{\mathsf{B}})$ *be the set containing rules and facts for deriving beliefs;* (Π_F,Δ_F)*, the set of filtering rules; and* $\Delta_X = \{(\alpha \prec \gamma) \mid (\alpha \leftarrow \gamma) \in (\Pi_{\mathsf{B}} \cup \Pi_F)$ *and* $(\Pi_{\mathsf{B}} \cup \Pi_F) \vdash \{\alpha, \overline{\alpha}\}\}$. *Then* $K_{Ag} = (\Pi_{\mathsf{B}} \circ \Pi_F, \Delta_{\mathsf{B}} \cup \Delta_F \cup \Delta_X)$ *will be the agent's knowledge base.*

The next definition introduces a mechanism for filtering D obtaining only those desires that are achievable in the *current* situation. We allow the representation of different *agent types*, each of which will specify a different filtering process.

Definition 4 (Current desires) *Let T be a selection criterion. The set D^c of* Current Desires *is defined as:* $\mathsf{D}^c = filter(T, \mathsf{D})$*, where the function $filter(\cdot,\cdot)$ returns the maximal subset of D containing those desires that satisfy the selection criterion T.*

Observe that the filtering function can be defined in a modular way. Methodologically, it would be important to make this function related to the K_{Ag}, in order to obtain a rational filtering. Implementing a sensible filtering function is not a trivial task, as it is domain-dependent, and a general criterion cannot be stated. Different agent types, personalities or behaviors can be obtained depending on the chosen filtering criterion. The following are interesting alternatives:
CAUTIOUS AGENT: $filter(T, \mathsf{D}) = \{\delta \in \mathsf{D} \mid T(\delta, K_{Ag}) =$ "there is a warrant for δ from K_{Ag}"$\}$
BOLD AGENT: $filter(T, \mathsf{D}) = \{\delta \in \mathsf{D} \mid T(\delta, K_{Ag}) =$ "there is no warrant for $\overline{\delta}$ from K_{Ag}"$\}$

Notice that when neither Q nor $\overline{Q}$ has a warrant built from K_{Ag}, then both literals will be included into the set D^c of a bold agent. Therefore, the agent will consider these two options (among others), albeit in contradiction.

The way a bold agent selects its current desires (see Ex. 4) becomes clearer considering the relation of warrant states with DeLP answers. In DeLP, given a literal Q, there are four possible answers for the query Q: YES, NO, UNDECIDED, and UNKNOWN. Thus, agent types using DeLP can be defined as follows:
CAUTIOUS AGENT: $filter(T, \mathsf{D}) = \{\delta \in \mathsf{D} \mid T(\delta, K_{Ag}) =$ "the answer for δ from K_{Ag} is YES"$\}$
BOLD AGENT: $filter(T, \mathsf{D}) = \{\delta \in \mathsf{D} \mid T(\delta, K_{Ag}) =$ "the answer for δ from K_{Ag} is YES, UNDECIDED or UNKNOWN"$\}$

Example 4 *Extending Ex. 3, if we consider a bold agent as defined above and the set of beliefs:*
$\mathsf{B} = \Phi = \{\ farFromGoal,\ noOneAhead,\ ball\ \}$
the agent will generate the following set of current desires:
$\mathsf{D}^c = \{carry, pass\}$
In this case, we have $K_{Ag} = (\Phi \circ \Pi_F, \emptyset \cup \Delta_F \cup \emptyset)$. Regarding D^c, DeLP's answer for shoot is NO*, for carry is* YES*, and for pass is* UNDECIDED*. Finally, note that a cautious agent would choose carry as the only current desire.*

As stated above, it is required that B and D be two separate sets to avoid the confusion when joining the (Π_{B},Δ_{B}) and (Π_F,Δ_F) programs. This is not a strong restriction, because a literal being both a belief and a desire brings about well-known representational issues.

Selecting Intentions

In our approach, an intention will be a current desire $d \in \mathsf{D}^c$ that the agent can commit. To specify under what conditions the intention could be achieved, the agent will be provided with a set of *intention rules*. Next, these concepts and the formal notion of *applicable intention rule* are introduced.

Definition 5 (Intention Rule) *An* intention rule *is a device used to specify under what conditions an intention could be effected. It will be denoted as $(d \Leftarrow \{p_1, \ldots, p_n\}, \{not\ c_1, \ldots, not\ c_m\})$, where d is a literal representing a desire that could be selected as an intention, $p_1, \ldots, p_n$ ($n \geq 0$) are literals representing preconditions, and $c_1, \ldots, c_m$ ($m \geq 0$) are literals representing constraints.*

Example 5 *A robotic-soccer agent might have the following set of intention rules:*
$IR_1: (carry \Leftarrow \{ball\}, \{\})$
$IR_2: (pass \Leftarrow \{ball\}, \{not\ shoot\})$
$IR_3: (shoot \Leftarrow \{ball\}, \{not\ marked\})$
$IR_4: (carry \Leftarrow \{winning\}, \{\})$
$IR_5: (move \Leftarrow \{\}, \{\})$

Now we describe how an intention becomes applicable.

Definition 6 (Applicable Intention Rule)
Let K_{Ag}= ($\Pi_{\mathsf{B}} \circ \Pi_F$, $\Delta_{\mathsf{B}} \cup \Delta_F \cup \Delta_X$) be the knowledge base of an agent, and D^c, its set of current desires. Let B be the set of beliefs obtained from (Π_{B},Δ_{B}). An intention rule $(d \Leftarrow \{p_1, \ldots, p_n\}, \{not\ c_1, \ldots, not\ c_m\})$ is applicable *iff*

1. *$d \in \mathsf{D}^c$,*
2. *for each precondition p_i ($0 \leq i \leq n$) it holds $p_i \in (\mathsf{B} \cup \mathsf{D}^c)$*
3. *for each constraint c_i ($0 \leq i \leq m$) it holds $c_j \notin (\mathsf{B} \cup \mathsf{D}^c)$.*

Thus, in every applicable intention rule it holds:

1. the head d is a current desire of the agent selected by the filtering function,
2. every precondition p_i that is a belief is warranted from K_{Ag},
3. every precondition p_i that is a desire belongs to set D^c,
4. every belief constraint c_i has no warrant from K_{Ag}, and
5. every c_i that is a desire does not belong to D^c.

Example 6 *Consider a bold agent, and K, B and D^c as given in Example 4. Now it is possible to determine which of the intention rules of Example 5 are applicable. Rule IR_1 is applicable because $carry \in \mathsf{D}^c$. Rule IR_2 is applicable because $pass \in \mathsf{D}^c$, $ball \in \mathsf{B}$, and $shoot \notin \mathsf{D}^c$. Rule IR_3 is not applicable because $shoot \notin \mathsf{D}^c$. Rule IR_4 is not applicable because the precondition is not a literal from K. Finally, IR_5 is not applicable because $move \notin \mathsf{D}^c$. Thus, $\{IR_1, IR_2\}$ is the set of applicable rules.*

Intention rules' goal is to select the final set of intentions. In general, this selection among current desires cannot be done by using filtering rules. For instance, if we have to select just one intention, and there are two warranted current desires, how can we choose one? There is a need for an external mechanism to make that decision.

Intention rules and filtering rules (Definition 2) have different semantics and usage:

- Filtering rules are used to build arguments for and against desires (thus, they are the basis of the dialectical process for warranting a desire), whereas intention rules are used on top of the dialectical process.
- Intention rules do not interact, whereas filtering rules do interact because they can be in conflict or can be used for deriving a literal in the body of another filtering rule.
- Applicable intention rules depend on the result of the filtering process over desires and warranted beliefs, whereas a filtering rule is "applicable" when its body literals are supported by perceived beliefs, or by other defeasible or strict rules.

The set of all applicable intention rules contains rules whose heads represent *applicable intentions* achievable in the current situation. Depending on the application domain, there are many possible policies to select from the set of applicable intentions. For example, the agent could try to pursue some of them simultaneously, or it might be forced to commit to one. Furthermore, each of these two options has, in turn, several solutions. The idea behind having intention rules and policies is to give a more flexible mechanism than plain priorities. Next, we define how to obtain a set of selected intentions.

Definition 7 (Set of Selected Intentions) *Let IR be the set of intention rules, and $App \subseteq IR$ be the set of all the applicable intention rules. Let $p : IR \rightarrow \mathsf{D}$ be a given selection policy. Then, the* set of selected intentions I *will be* $p(App)$.

The policy $p(App)$ could be defined in many ways. For instance, $p(App)$ could be "return all the heads of rules in App". However, depending on the application domain, more restrictive definitions for $p(App)$ could be necessary. For example, in our robotic soccer domain, agents must select a single applicable intention at a time (*i.e.*, an agent cannot shoot and pass the ball at the same time). One possibility for defining a policy that returns a single intention is to provide a sequence with all the intention rules $[IR_1,...,IR_n]$ that represents a preference order among them. Then, the policy $p(App)$ selects the first rule IR_k $(1 \leq k \leq n)$ in the sequence that belongs to App, returning the head of IR_k.

Example 7 *Continuing with Ex. 6. The set of applicable intention rules is $App = \{IR_1, IR_2, IR_5\}$, and suppose that the policy p is the one introduced above. Then, if the preference order is $[IR_1, IR_2, IR_3, IR_4, IR_5]$, the selected intention will be the head of IR_1, i.e., $p(App) = \{carry\}$.*

Now we can formally define the structure of an agent.

Definition 8 (DeLP-Based BDI Agent)
An agent A is a tuple $\langle \mathsf{D}, (\Pi_\mathsf{B},\Delta_\mathsf{B}), (\Pi_F,\Delta_F), T, IR, p\rangle$, where: D is the set of desires of the agent, $(\Pi_\mathsf{B},\Delta_\mathsf{B})$ is the agent knowledge (that will include perceived beliefs), (Π_F,Δ_F) are filtering rules, T is an agent type, IR is a set of intention rules, and $p(\cdot)$ is a policy for selecting intentions.

We will show, using different examples, how the proposed agent selects appropriate intentions when faced with different scenarios. In each example, the difference of defining a bold or a cautious agent will be made clear.

Example 8 *Let $A{=}\langle \mathsf{D}, (\Pi_\mathsf{B},\Delta_\mathsf{B}), (\Pi_F,\Delta_F), T, IR, p\rangle$ be an agent with the set D and (Π_F,Δ_F) from Ex. 3, the set IR of Ex. 5, the policy p defined in Ex. 7, and the set $\Delta_\mathsf{B} = \emptyset$. Consider the situation in Fig. 2(a) where "o1" and "o2" represent the positions of two opponents and "self" is the position of the agent A who has the ball (small circle). Thus, the perception of the agent is $\Phi_1 = \{ball, noOneAhead, theirGoalieAway\}$. In this situation, agent A can build the following arguments:*

$\mathcal{A}_1 : \{shoot \prec theirGoalieAway\}$,
$\mathcal{A}_2 : \{carry \prec noOneAhead\}$,
$\mathcal{A}_3 : \{(\sim carry \prec shoot), (shoot \prec theirGoalieAway)\}$.

*Hence, shoot is warranted, whereas carry, $\sim$carry, pass and $\sim$pass are not. As stated above the filter function will determine the type of agent (*e.g., *bold or cautious), which could affect the set of selected intentions. For example:*

- *for a cautious agent, $\mathsf{D}^c_{C1} = \{shoot\}$, intention rule IR_3 is applicable, and $\mathsf{I}_{C1} = \{shoot\}$;*
- *for a bold agent, $\mathsf{D}^c_{B1} = \{shoot, carry, pass\}$, intention rules IR_1 and IR_3 are applicable, and $\mathsf{I}_{B1} = \{carry\}$.*

Note that the cautious agent obtains only one current desire that is its selected intention. On the other hand, since the bold agent includes "undecided" literals in its current desires, D^c_{B1} has more elements than D^c_{C1}, there are two applicable intention rules, and the policy "p" has to be used.

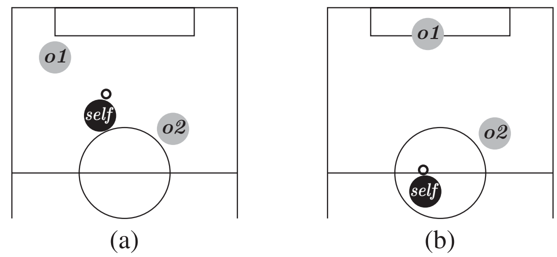

Figure 2: Two scenarios for a robotic soccer agent

Example 9 *Consider the agent A in Ex. 8 but in a different scenario (depicted in Fig. 2(b)). Here the perception of the agent is $\Phi_2 = \{ball, noOneAhead, farFromGoal\}$.*
In this situation, agent A can build the following arguments:

$\mathcal{A}_1 : \{\sim shoot \prec farFromGoal\}$,
$\mathcal{A}_2 : \{carry \prec noOneAhead\}$.

Hence, $\sim$shoot and carry are warranted, whereas pass and $\sim$pass are not, and:

- *for a cautious agent, $\mathsf{D}^c_{C2} = \{carry\}$, intention rule IR_1 is applicable, and $\mathsf{I}_{C2} = \{carry\}$;*
- *for a bold agent, $\mathsf{D}^c_{B2} = \{carry, pass\}$, intention rules IR_1 and IR_2 are applicable, and $\mathsf{I}_{B2} = \{carry\}$.*

Example 10 *Consider the situation depicted in Fig. 3(a) for the agent A in Ex. 8, where "t1" represents the position of a teammate of A. The perception of A is $\Phi_3 = \{ball, freeTeammate, farFromGoal\}$. In this situation, A can build the following arguments:*

$\mathcal{A}_1 : \{\sim shoot \prec farFromGoal\}$,
$\mathcal{A}_2 : \{pass \prec freeTeammate\}$,

Hence, we have that pass and $\sim$shoot are warranted, whereas carry and $\sim$carry are not, and:

- *for a cautious agent, $\mathsf{D}^c_{C3} = \{pass\}$, intention rule IR_2 is applicable, and $\mathsf{I}_{C3} = \{pass\}$;*
- *for a bold agent, $\mathsf{D}^c_{B3} = \{carry, pass\}$, intention rules IR_1 and IR_2 are applicable, and $\mathsf{I}_{B3} = \{carry\}$;*

Now, in the situation of (Fig. 3(b)) for the agent A of Ex. 8. The perception of the agent is $\Phi_4 = \{ball, freeTeammate, theirGoalieAway\}$. We can build the following arguments:

$\mathcal{A}_1 : \{shoot \prec theirGoalieAway\}$,
$\mathcal{A}_2 : \{pass \prec freeTeammate\}$,
$\mathcal{A}_3 : \{(\sim carry \prec shoot), (shoot \prec theirGoalieAway)\}$.

Hence, pass, shoot and $\sim$carry are warranted, and:

- *for a cautious agent, $\mathsf{D}^c_{C4} = \{shoot\}$, intention rule IR_3 is applicable, and $\mathsf{I}_{C4} = \{shoot\}$;*
- *for a bold agent, $\mathsf{D}^c_{B4} = \{shoot\}$, intention rules IR_3 is applicable, and $\mathsf{I}_{B4} = \{shoot\}$.*

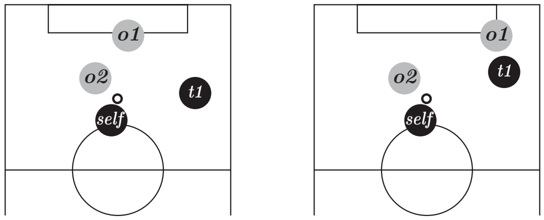

Figure 3: Two scenarios for a robotic soccer agent

Related Work

The use of defeasible argumentation in BDI architectures is not new and it was originally suggested in (Bratman, Israel, & Pollack 1991), and more recently in (Parsons, Sierra, & Jennings 1998). Also in (Thomason 2000) and (Broersen *et al.* 2001) a formalism for reasoning about beliefs and desires is given, but they do not use argumentation.

Recently, Rahwan and Amgoud (2006) have proposed an argumentation-based approach for practical reasoning that extends (Amgoud 2003) and (Amgoud & Cayrol 2002), introducing three different instantiations of Dung's framework to reason about beliefs, desires and plans, respectively. This work is, in our view, the one most related to ours. Both approaches use defeasible argumentation 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. Both approaches construct arguments supporting competing desires, and they are compared and evaluated to decide which one prevails. Their notion of *desire rule* is similar to our *filtering rules*.

In their approach, two different argumentation frameworks are needed to reason about desires: one framework for beliefs rules and another 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 explicit mention of perceived information is given. In our case, uncertainty is represented by defeasible rules (García & Simari 2004) and perceived beliefs are explicitly treated by the model. Besides, the argumentation formalism used in their approach differs from ours: their comparison of arguments relies on the certainty factor given to each formula, and they do not distinguish between proper and blocking defeaters. Another fundamental difference is that we permit the definition of different types of agents. This feature adds great flexibility in the construction of an agent.

Conclusions

We have shown how a deliberative agent can represent its perception and beliefs using a defeasible logic program. The information perceived directly from the environment is represented with a subset of perceived beliefs that is dynamically updated, and a set formed with strict rules and facts represent other static knowledge of the agent. In addition to this, defeasible argumentation is used to warrant agents (derived) beliefs. Strict and defeasible filtering rules have been introduced to represent knowledge regarding desires. Defeasible argumentation is 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 and select the appropriate ones. We allow the representation of different agent types, each of which will specify a different way to perform the filtering process. In our approach, an intention is a current desire that the agent can commit to pursue. The agent is provided with a set of intention rules that specify under what conditions an intention could be achieved. If there is more than one applicable intention rule, then a policy is used to define a preference criterion among them. Thus, intention policies give the agent a mechanism for deciding which intentions should be selected in the current situation.

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 *Proc. of the 7th ECSQARU*, 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, MA: The MIT Press. 1–22.

Broersen, J.; Dastani, M.; Hulstijn, J.; Huang, Z.; and van der Torre, L. 2001. The boid architecture: conficts between beliefs, obligations, intentions and desires. In *Proc. of 5th Int. Conf. on Autonomous Agents*. Canada: ACM Press. 9–16.

Falappa, M.; Kern-Isberner, G.; and Simari, G. 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, CA.

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

Hulstijn, J., and van der Torre, L. 2004. Combining goal generation and Planning in and argumentation framework. In *Proc. of the 10th NMR*.

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.

Rahwan, I., and Amgoud, L. 2006. An argumentation-based approach for practical reasoning. In *Proc. of the 5th AAMAS*.

Rotstein, N., and García, A. 2006. Defeasible reasoning about beliefs and desires. In *Proc. of the 11th NMR*, 429–436.

Thomason, R. 2000. Desires and defaults: A framework for planning with inferred goals. In *Proc. of the seventh KR*, 702–713.