Preferences of Agents in Defeasible Logic*

Mehdi Dastani[1], Guido Governatori[2], Antonino Rotolo[3], and Leendert van der Torre[4]

[1] Intelligent Systems Group, Utrecht University,
P.O.Box 80.089, NL-3508 TB Utrecht, The Netherlands, Email: mehdi@cs.uu.nl
[2] School of ITEE, The University of Queensland
Brisbane QLD 4072, Australia, Email: guido@itee.uq.edu.au
[3] CIRSFID, University of Bologna
Via Galliera 3, I-40121 Bologna, Italy, Email: rotolo@cirsfid.unibo.it
[4] CWI, Amsterdam, and Delft University of Technology
Kruislaan 413, NL-1098 SJ Amsterdam, The Netherlands, Email: torre@cwi.nl

Abstract. We are interested in programming languages for cognitive agents with preferences. We define rule-based agent theories and inference procedures in defeasible logic, and in this setting we discuss patterns of agent behavior called agent types.

1 Introduction

There are several rule-based approaches to programming cognitive agents [4, 7, 2]. In this paper we extend the Defeasible Logic (DL) approach. DL is a simple, efficient but flexible non-monotonic formalism able to deal with many different intuitions of non-monotonic reasoning and recently applied in many fields. Here we propose a non-monotonic logic of agency which extends the framework developed in [1]. DL is one of the most expressive languages of the type we are interested in, and in particular it has defined the largest set of patterns called agent types. Moreover, it is flexible to incorporate ideas from other languages which have not been introduced yet, such as extension generation and selection from BOID [4], or deliberation languages from 3APL [7, 5].

However, it has two drawbacks. First, DL, as well as its rival rule based programming languages, is based on a uniform representation of rules, whereas in artificial intelligence and in practical reasoning other complex structures have been proposed. Most importantly, rule-based approaches are based on conditionals, whereas an alternative approach is based on comparative notions. Examples are preference logics and CP nets instead of logics of desires and goals, ordered disjunctions instead of default logics, betterness logics instead of logics of ideality, logics of sub-ideality in deontic logic, etc. Second, it is not immediate how DL can deal with complex actions discussed in action languages such as 3APL [7] and in recent incarnations of the BOID architecture [6]. In this paper we address the first drawback while the second is left for future research.

Some issues of agent programming languages should be addressed: how to detect and resolve conflicts that include such preferences, and which kind of agent types can be

* This work was partially supported by Australia Research Council under Discovery Project No. DP0558854 on "A Formal Approach to Resource Allocation in Web Service Oriented Composition in Open Marketplaces".

introduced to deal with preferences. Summarizing, we therefore contribute to cognitive agent programming languages by addressing the following research question: *How to use DL extended with graded preferences?* This question breaks down in the following sub-questions:

1. How to introduce graded preferences in DL?
2. How to detect and resolve conflicts using preferences?
3. How to define agent types based on preferences?

For the representation of preferences we use a variant of the $\otimes$ operator of [8] in DL, because it has several advantages over other comparative notions. First, it has already been integrated with a rule based formalism. Second, it has been applied to complicated problems in deontic logic, e.g., to the so-called contrary-to-duty reasoning. Third, it allows to clearly distinguish between conflicts and violations within a rule-based system. In fact, though these notions may conflate, conflicts and violations have in general to be kept separate. Suppose you have an agent doing A while an obligation states $\mathrm{OBL}\neg A$. Since the logic for OBL is usually not reflexive, the scenario does not lead necessarily to a logical conflict but a violation: indeed, conflict-resolution strategies may require that $\mathrm{OBL}\neg A$ is not overriden.

In this paper we focus on goal generation based on beliefs, desires, intentions and obligations. Our system is rule-based, and rules will allow to derive new motivational factors of an agent. We will divide the rules into rules for beliefs, desires, intentions, and obligations. Provability for beliefs will not generate goals, since in our view they concern agent's knowledge about the world: beliefs may contribute to derive goals (desires, intentions, and obligations), but they are not in themselves motivations for action.

The layout of this paper is as follows. In Section 2 we introduce agents with preferences in DL, and in Section 3 we show how to infer goal conclusions from rules with preferences. In Section 4 we discuss conflicts among rules and patterns called agent types.

2 Agents in defeasible logic

We focus on how mental attitudes and obligations jointly interplay in modeling agent deliberation and behavior. The formal language contains modal literals, and preferences (including literals as a borderline case).

Definition 1 (Language). *Let $M = \{\mathrm{BEL}, \mathrm{DES}, \mathrm{INT}, \mathrm{OBL}\}$ be a set of modal operators, and P a set of propositional atoms.*

- *The set of literals is defined as $L = P \cup \{\neg p | p \in P\}$.*
- *If q is a literal, $\sim q$ denotes the complementary literal (if q is a positive literal p then $\sim q$ is $\neg p$; and if q is $\neg p$, then $\sim q$ is p).*
- *Given the set of literals and the modal operators* BEL, DES, INT, *and* OBL, *if l is a literal and X is a modal operator, then Xl and $\neg Xl$ are modal literals. Every literal l is an $\otimes$-expression. If $l_1, \ldots, l_n$ are literals then $l_1 \otimes \ldots \otimes l_n$ is an $\otimes$-expression. The goal language L_{goal} is the smallest set that contains literals, modal literals and $\otimes$-expressions.*

For $X \in \{\text{BEL}, \text{INT}, \text{DES}, \text{OBL}\}$, we have that $\phi \rightarrow_X \psi$ is a *strict rule* such that whenever the premises ϕ are indisputable so is the conclusion ψ. $\phi \Rightarrow_X \psi$ is a *defeasible rule* that can be defeated by contrary evidence. $\phi \leadsto_X \psi$ is a *defeater* that is used to defeat some defeasible rules by producing evidence to the contrary.

Definition 2 (Rules). *A* rule *r consists of its* antecedent *(or* body*) $A(r)$ ($A(r)$ may be omitted if it is the empty set), an arrow ($\rightarrow$ for a strict rule, $\Rightarrow$ for a defeasible rule, and $\leadsto$ for a defeater), and its* consequent *(or* head*) $C(r)$. In addition the arrow is labelled either with a modal operator $X \in M$. If the arrow is labelled with* BEL *the rule is for belief, and similarly for the other modal operators.*

– *Given a rule r, $A(r)$ is a set of literals or modal literals, and $C(r)$ is a literal with no occurrence of $\otimes$ for strict rules, an $\otimes$-expression for defeasible rules and defeaters.*
– *Given a set R of rules, we denote the set of all strict rules in R by R_s, the set of strict and defeasible rules in R by R_{sd}, the set of defeasible rules in R by R_d, and the set of defeaters in R by R_{dft}. $R[q]$ denotes the set of rules in R with consequent q. For some i, $1 \leq i \leq n$, such that $c_i = q$, $R[c_i = q]$ and $r_d^X[c_i = q]$ denote, respectively, the set of rules and a defeasible rule of type X with the head $\otimes_{i=1}^n c_i$.*

Rules for beliefs are meant to constitute the reasoning core of the system. The purpose of goal generation is to derive the other modal literals. For example, the application of $p \Rightarrow_{\text{INT}} q$ permits to infer INTq. Accordingly, modalities do not occur in the consequents of rules to keep the system manageable.

Definition 3 (Defeasible agent theory). *A defeasible agent theory is a structure $D = (F, R^{\text{BEL}}, R^{\text{DES}}, R^{\text{INT}}, R^{\text{OBL}}, >)$ where F is a finite set of facts, R^{BEL} is a finite set of rules for belief, R^{DES} is a finite set of rules for desire, R^{INT} is a finite set of rules for intention, R^{OBL} is a finite set of rules for obligation, and $>$, the superiority relation, is a binary relation over the set of rules.*

The *superiority relation* $>$ says when one rule may override the conclusion of another rule. *Facts* are indisputable statements. Beside the superiority relation, which is used when we have contradictory conclusions, we can establish a preference over the conclusions by using the operator $\otimes$. Thus, the intuitive reading of a sequence like $a \otimes b \otimes c$ means that a is preferred, but if $\neg a$ is the case, then b is preferred; if $\neg b$ is the case, then the third choice is c.[5]

Definition 4 (Operators). *A preference operator $\otimes$ is a binary operator satisfying the following properties: (1) $a \otimes (b \otimes c) = (a \otimes b) \otimes c$ (associativity); (2) $\bigotimes_{i=1}^n a_i = (\bigotimes_{i=1}^{k-1} a_i) \otimes (\bigotimes_{i=k+1}^n a_i)$ where exists j such that $a_j = a_k$ and $j < k$ (duplication and contraction on the right).*

The general idea of degree of preferences is that $\otimes$ formulas are interpreted as preference formulas like in [8] and are here extended to cover all motivational component: *for beliefs*, a construction such as $\neg$*SunShining* $\Rightarrow_{\text{BEL}}$ *Raining* $\otimes$ *Snowing* says

[5] A similar approach, but with a different motivation has been proposed in the context of logic programming by Brewka and co-worker in their logic of ordered disjunction [3].

that the agent believes that it is raining, but if it is not raining then it is snowing as the sun is not shining; *for desires*, rule $\mathit{TimeForHoliday} \Rightarrow_{\mathrm{DES}} \mathit{GoToAustralia} \otimes \mathit{GoToSpain}$ means that, if it is time for holiday, the agent has the primary desire to go to Australia, but, if this is not the case, her desire is to go to Spain; *for intentions*, rule $\mathit{SunShining} \Rightarrow_{\mathrm{INT}} \mathit{Jogging} \otimes \mathit{Walking}$ says that the agent intends to do jogging if the sun is shining, but, if, for some other reasons, this is not the case, then she will have the intention to have a walk; *for obligations*, rule $\mathit{Order} \Rightarrow_{\mathrm{OBL}} \mathit{Pay} \otimes \mathit{PayInterest}$ says that, if the agent sends her purchase order, then she will be obliged to pay, but, in the event this is not done, she will have to pay interest.

The $\otimes$ formulas are characterized by the following subsumption relation among rules, which is used in the following section to infer goals from an agent theory.

Definition 5. *Let* $r_1 = \Gamma \Rightarrow_X \bigotimes_{i=1}^{m} a_i \otimes b$ *and* $r_2 = \Gamma' \Rightarrow_X c$ *be two goal rules. Then* r_1 subsumes r_2 *iff* $\Gamma \cup \{\neg a_1, \ldots, \neg a_m\} = \Gamma'$, $c = e$ *and* $b = (e \otimes (\bigotimes_{k=1}^{n} d_k))$, $0 \leq n$.

The following example illustrates subsumption.

Example 1. $\mathit{Order} \Rightarrow_{\mathrm{OBL}} \mathit{Pay} \otimes \mathit{PayInterest}$ subsumes $\mathit{Order} \Rightarrow_{\mathrm{OBL}} \mathit{Pay}$. Moreover, $\mathit{Order} \Rightarrow_{\mathrm{OBL}} \mathit{Pay} \otimes \mathit{PayInterest}$ subsumes $\mathit{Order}, \neg \mathit{Pay} \Rightarrow_{OBL} \mathit{PayInterest}$.

The following example illustrates the agent theory.

Example 2. (Running example) Suppose an agent desires an application server. She can buy two products from X or Y. In general she prefers X but, for working with Linux, she does not intend to order X's product. X requires a payment, within 2 days, of 300\$, otherwise X forbids to download the software. Y requires a payment of 600\$ within 1 day, or, as a second choice, a payment of 660\$. The agent, for some reasons, does not intend to pay to Y 660\$. Agent's financial resources amount to 700\$, which are available in 4 days. As facts, we also know that the agent is a Linux user. The following piece of theory is considered to derive the agent's goals.

$$\begin{aligned}
&F = \{\mathit{700\$In4days}, \mathit{UseLinux}, \mathrm{DES}\mathit{Applserver}\}\\
&R = \{r_1 : \mathit{700\$In4days} \Rightarrow_{\mathrm{BEL}} \neg \mathit{PayY600\$1days},\ r_2 : \mathit{700\$In4days} \Rightarrow_{\mathrm{BEL}} \neg \mathit{PayX300\$2days},\\
&\qquad r_3 : \mathrm{DES}\mathit{Applserver} \Rightarrow_{\mathrm{INT}} \mathit{OrderX} \otimes \mathit{OrderY},\ r_4 : \mathit{UseLinux} \Rightarrow_{\mathrm{INT}} \neg \mathit{OrderX}\\
&\qquad r_5 : \mathrm{INT}\mathit{OrderY} \Rightarrow_{\mathrm{INT}} \neg \mathit{PayY660\$},\ r_6 : \mathrm{INT}\mathit{OrderY} \Rightarrow_{\mathrm{OBL}} \mathit{PayY600\$1days} \otimes \mathit{PayY660\$},\\
&\qquad r_7 : \mathrm{INT}\mathit{OrderX} \Rightarrow_{\mathrm{OBL}} \mathit{PayX300\$2days} \otimes \neg \mathit{DownloadApplserverX}\}\\
&>= \{r_4 > r_3\}
\end{aligned}$$

3 Goal generation: inference with preferences

Proofs are based on proof tags $+\Delta$, $-\Delta$, $+\partial$ and $-\partial$. $+\Delta_X q$ means that q is provable using only facts and strict rules for X, $-\Delta_X q$ means that it has been proved that q is not definitely provable, $+\partial_X q$ means that q is defeasibly provable in D and $-\partial_X q$ means that it has been proved that q is not defeasibly provable.

Definition 6 (Proofs). *Given an agent theory D, a proof in D is a linear derivation, i.e, a sequence of labelled formulas of the type* $+\Delta_X q$, $-\Delta_X q$, $+\partial_X q$ *and* $-\partial_X q$, *where the proof conditions defined in the rest of this section hold.*

We start with some terminology. As explained in the previous section, the following definition states the special status of belief rules, and that an introduction of a modal operator corresponds to being able to derive the associated literal using the rules for the modal operator.

Definition 7. *Let* $\# \in \{\Delta, \partial\}$*,* $X \in \{\mathrm{DES}, \mathrm{INTOBL}\}$*, and* $P = (P(1), \ldots, P(n))$ *be a proof in D. A literal q is* #-provable *in P if there is a line* $P(m)$ *of P such that either*

1. *q is a literal and* $P(m) = +\#_{\mathrm{BEL}} q$ *or*
2. *q is a modal literal* Xp *and* $P(m) = +\#_X p$ *or*
3. *q is a modal literal* $\neg Xp$ *and* $P(m) = -\#_X p$*;*

a literal q is #-rejected *in P if there is a line* $P(m)$ *of P such that either*

1. *q is a literal and* $P(m) = -\#_{\mathrm{BEL}} q$ *or*
2. *q is a modal literal* Xp *and* $P(m) = -\#_X p$ *or*
3. *q is a modal literal* $\neg Xp$ *and* $P(m) = +\#_X p$*.*

The first type of tagged literals, denoted by Δ_X, correspond to strict rules. The definition of Δ_X describes just forward chaining of strict rules:

$+\Delta_X$: If $P(i+1) = +\Delta_X q$ then
(1) $q \in F$ or
(2) $\exists r \in R_s^X[q]\ \forall a \in A(r)\ a$ is Δ-provable or
(3) $\exists r \in R_s^{\mathrm{BEL}}[q]\ \forall a \in A(r)\ Xa$ is Δ-provable.

$-\Delta_X$: If $P(i+1) = -\Delta_X q$ then
(1) $q \notin F$ and
(2) $\forall r \in R_s^X[q]\ \exists a \in A(r) : a$ is Δ-rejected and
(3) $\forall r \in R_s^{\mathrm{BEL}}[q]\ \exists a \in A(r)\ Xa$ is Δ-rejected.

For a literal q to be definitely provable we need to find a strict rule with head q, whose antecedents have all been definitely proved previously. And to establish that q cannot be proven definitely we must establish that for every strict rule with head q there is at least one of antecedent which has been shown to be non-provable. Condition (3) says that a belief rule can be used as a rule for a different modal operator in case all literals in the body of the rules are modalized with the modal operator we want to prove.

Conditions for ∂_X are more complicated since we have to consider $\otimes$-expressions that may occur in defeasible rules. We define when a rule is applicable or discarded. A rule for a belief is applicable if all the literals in the antecedent of the rule are provable with the appropriate modalities, while the rule is discarded if at least one the literals in the antecedent is not provable. For the other types of rules we have to take complex derivations into account called conversions [9]. We say there is a conversion from X to Y if a X rule can also be used as a Y rule. We have thus to determine conditions under which a rule for X can be used to directly derive a literal q modalized by Y. Roughly, the condition is that all the antecedents a of the rule are such that $+\partial_Y a$. We represent all allowed conversions by a conversion relation c (see next section for further interpretation of conversions in terms of agent types).

Definition 8. *Let a conversion relation c be a binary relation between* $\{\mathrm{BEL}, \mathrm{INT}, \mathrm{DES}, \mathrm{OBL}\}$*, such that* $c(X,Y)$ *stands for the conversion of X rules into Y rules.*

– *A rule r in* R^{BEL} *is* applicable *iff* $\forall a \in A(r)$*,* $+\partial_{\mathrm{BEL}} a \in P(1..n)$ *and* $\forall Xa \in A(r)$*, where X is a modal operator,* $+\partial_X a \in P(1..n)$*.*

- *A rule $r \in R_{sd}[c_i = q]$ is applicable in the condition for $\pm\partial_X$ iff*
  1. *$r \in R^X$ and $\forall a \in A(r)$, $+\partial a \in P(1..n)$ and $\forall Ya \in A(r)$ $+\partial_Y a \in P(1..n)$, or*
  2. *$r \in R^Y$ and $\forall a \in A(r)$, $+\partial_X a \in P(1..n)$.*
- *A rule r is discarded if we prove either $-\partial_{\text{BEL}} a$ or $-\partial_X a$ for some $a \in A(r)$.*

Example 3. The belief rule $a, \text{INT}b \Rightarrow_{\text{BEL}} c$ is applicable if we can prove $+\partial_{\text{BEL}} a$ and $+\partial_{\text{INT}} b$.

Example 4. If we have a type of agent that allows a deontic rule to be converted into a rule for intention, $c(\text{OBL}, \text{INT})$, then the definition of applicable in the condition for $\pm\partial_{\text{INT}}$ is as follows: a rule $r \in R_{sd}[c_i = q]$ is applicable iff (1) $r \in R^{\text{INT}}$ and $\forall a \in A(r)$, $+\partial a \in P(1..n)$ and $\forall Xa \in A(r)$, $+\partial_X a \in P(1..n)$, (2) or $r \in R^O$ and $\forall a \in A(r)$, $+\partial_{\text{INT}} a \in P(1..n)$.

The second type of tagged literals, denoted by ∂, correspond to defeasible rules. Two cases of these tagged literals are distinguished: $+\partial_X$ to indicate positive defeasible provability for the modality X and $-\partial_X$ to indicate negative defeasible provability for the modality X. Proof conditions for $\pm\partial_X$ are thus as follows:

$+\partial_X$: If $P(n+1) = +\partial_X q$ then
(1)$+\Delta_X q \in P(1..n)$ or
  (2.1) $-\Delta_X {\sim} q \in P(1..n)$ and
  (2.2) $\exists r \in R_{sd}[c_i = q]$ such that r is applicable, and
    $\forall i' < i$, $-\partial_{\text{BEL}} c_{i'} \in P(1..n)$; and
  (2.3) $\forall s \in R[c_j = {\sim} q]$, either s is discarded, or
    $\exists j' < j$ such that $+\partial_X c_{j'} \in P(1..n)$, or
  (2.3.1) $\exists t \in R[c_k = q]$ s.t. r is applicable and
    $\forall k' < k$, $-\partial_{\text{BEL}} c_{k'} \in P(1..n)$ and $t > s$

$-\partial_X$: If $P(n+1) = -\partial_X q$ then
(1) $-\Delta_X q \in P(1..n))$ and either
  (2.1) $+\Delta_X {\sim} q \in P(1..n)$ or
  (2.2) $\forall r \in R_{sd}[c_i = q]$, either r is discarded or
    $\exists i' < i$ such that $+\partial_{\text{BEL}} c_{i'} \in P(1..n)$, or
  (2.3) $\exists s \in R[c_j = {\sim} q]$, such that s is applicable and
    $\forall j' < j$, $-\partial_X c_{j'} \in P(1..n)$ and
  (2.3.1) $\forall t \in R[c_k = q]$ either t is discarded, or
    $\exists k' < k$ such that $+\partial_{\text{BEL}} c_{k'} \in P(1..n)$ or $t \not> s$

For defeasible rules we deal with $\otimes$ formulas, and this is where the subsumption relation comes into the system. Roughly, a rule $a \Rightarrow_X b \otimes c$ is interpreted as two rules $a \Rightarrow_X b$ and $a, \neg b \Rightarrow_X c$. To show that q is provable defeasibly we have two choices: (1) We show that q is already definitely provable; or (2) we need to argue using the defeasible part of D. For this second case, three (sub)conditions must be satisfied. First, we require that there must be a strict or defeasible rule for q which can be applied (2.1). Second, we need to consider possible reasoning chains in support of ${\sim}q$, and show that ${\sim}q$ is not definitely provable (2.2). Third, we must consider the set of all rules which are not known to be inapplicable and which permit to get $\sim q$ (2.3). Essentially each such a rule s attacks the conclusion q. For q to be provable, s must be counterattacked by a rule t for q with the following properties: (i) t must be applicable, and (ii) t must be stronger than s. Thus each attack on the conclusion q must be counterattacked by a stronger rule. In other words, r and the rules t form a team (for q) that defeats the rules s. $-\partial_X q$ is defined in an analogous manner.

The purpose of the $-\partial_X$ inference rules is to establish that it is not possible to prove $+\partial_X$. This rule is defined in such a way that all the possibilities for proving $+\partial_X q$ (for example) are explored and shown to fail before $-\partial_X q$ can be concluded. Thus conclusions tagged with $-\partial_X$ are the outcome of a constructive proof that the corresponding positive conclusion cannot be obtained.

Goals are obtained as $+\partial_G$ or $+\Delta_G$, $G \in \{\mathrm{DES}, \mathrm{INT}, \mathrm{OBL}\}$. As it was said, provability for beliefs does not generate goals, since beliefs concern agent's knowledge about the world.

Example 5. (Running example; continued) Let us assume that the agent is realistic, namely that beliefs override all motivational components (see Section 4). Below is the set C of all conclusions we get using the rules in R:

$$C = \{\neg \mathit{PayY600\$1days},\ \neg \mathit{PayX300\$2days},\ \mathrm{INT}\,\mathit{OrderY},\\ \mathrm{INT}\neg \mathit{OrderX},\ \mathrm{INT}\neg \mathit{PayY660\$}\}$$

Since the agent desires an application server, from r_3, r_4, $r_4 > r_3$ and $\otimes$-elimination, we have $+\partial_{\mathrm{INT}} \mathit{OrderY}$. This makes r_6 and r_5 applicable, while r_7 is not. However, the agent will have 700 \$ available within 4 days and so, since the agent is realistic, from r_1 we get $+\partial_{\mathrm{BEL}}\neg \mathit{PayY600\$1days}$, which is a violation of the primary obligation in r_6. We would obtain $+\partial_{\mathrm{OBL}} \mathit{PayY660\$}$, but this not the case since the theory does not provide criteria for resolving the conflict between this conclusion and that of r_5.

4 Goal generation: conflict resolution and agent types

Classically, agent types are characterized by stating conflict resolution types in terms of orders of overruling between rules [4, 9]. For example, an agent is *realistic* when rules for beliefs override all other components; she is *social* when obligations are stronger than the other components with the exception of beliefs. Agent types can be characterized by stating that, for any types of rules X and Y, for every r and r' such that $r \in R^X[c_i = q]$ and $r' \in R^Y[d_i = \sim q]$, we have that $r > r'$.

Let us assume to work with realistic agents, namely, with agents for which, for every r and r', $r \in R^{\mathrm{BEL}}[c_i = q]$ and $r' \in R^Y[d_i = \sim q]$, $Y \in \{\mathrm{DES}, \mathrm{INT}, \mathrm{OBL}\}$ we have that $r > r'$. Table 1 shows al possible cases and, for each kind of rule, indicates all attacks on it. Since we have defined four kinds of rules for generating goals, we have to analyze twelve combinations. (To save space, in Table 1 "s-" is an abbreviation for "strongly-".) Independent and strongly-independent agents are free respectively to adopt desires and intentions in conflict with obligations. For social and strongly-social agents obligations override desires and intentions. For pragmatic and strongly-pragmatic, no derivation is possible and so the agent's generation of goals is open to any other course of action

$r_d^{\mathrm{OBL}}[c_i = p]/\, r_d^{\mathrm{INT}}[c_j = \sim p]$			$r_d^{\mathrm{OBL}}[c_i = p]/\, r_d^{\mathrm{DES}}[c_j = \sim p]$			$r_d^{\mathrm{INT}}[c_i = p]/\, r_d^{\mathrm{DES}}[c_j = \sim p]$		
$+\partial_{\mathrm{OBL}} p$	$+\partial_{\mathrm{INT}} {\sim} p$	s-independent	$+\partial_{\mathrm{OBL}} p$	$+\partial_{\mathrm{DES}} {\sim} p$	independent	$+\partial_{\mathrm{INT}} p$	$+\partial_{\mathrm{DES}} {\sim} p$	unstable
$+\partial_{\mathrm{OBL}} p$	$-\partial_{\mathrm{INT}} {\sim} p$	s-social	$+\partial_{\mathrm{OBL}} p$	$-\partial_{\mathrm{DES}} {\sim} p$	social	$+\partial_{\mathrm{INT}} p$	$-\partial_{\mathrm{DES}} p$	stable
$-\partial_{\mathrm{OBL}} p$	$+\partial_{\mathrm{INT}} {\sim} p$	s-deviant	$-\partial_{\mathrm{OBL}} p$	$+\partial_{\mathrm{DES}} {\sim} p$	deviant	$-\partial_{\mathrm{INT}} p$	$+\partial_{\mathrm{DES}} {\sim} p$	selfish
$-\partial_{\mathrm{OBL}} p$	$-\partial_{\mathrm{INT}} {\sim} p$	s-pragmatic	$-\partial_{\mathrm{OBL}} p$	$-\partial_{\mathrm{DES}} {\sim} p$	pragmatic	$-\partial_{\mathrm{INT}} p$	$-\partial_{\mathrm{DES}} {\sim} p$	slothful

Table 1. Agent Types: Attacks

other than those specified in the rules considered. Stable and selfish agents are those for which, respectively, intentions override desires or the opposite. Unstable agents are free to adopt desires in conflict with intentions, while, for slothful agents, conflicting desires and intentions override each other[6].

This taxonomy can be enriched thanks to the role played by $\otimes$-expressions. In fact, in traditional rules-based systems, conflict-detection returns a boolean: either there is a conflict, or there is not. For $\otimes$ constructs, it seems that we may need a finer distinction. For example, we can have degrees of violation. Of course, if we define a conflict detection function that returns no longer booleans but a more complex structure (e.g., an integer that returns 0 if no violation, 1 if violation of primary obligation, 2 if violation of secondary obligation), then we have to write conflict resolution methods which can somehow deal with this. Section 3 provides criteria to solve conflict between rules including $\otimes$ constructions. In this perspective, the role of $\otimes$ can be made fruitful. In particular, the introduction of $\otimes$ is crucial if we want to impose some constraints on the number of violations in deriving a goals. Goal generation can be constrained, so that provability of a goal g is permitted only if getting g does not require more than n violations for each rule with g in the head:

Definition 9 (Violation constraint on goals). *Let n and X be an integer and a type of rule, respectively. A theory D will be n-X-constrained iff, given the definition of $+\partial$, for all literals q, $+\partial_X q$ iff (1) $i'' \leq n$; and (2) if $1 \leq j'' \leq j'$ and $s \in R_X$, then $j'' \leq n$; and (3) $k' \leq n$. Otherwise, $-\partial_X q$.*

Similar intuitions are applicable to directly constraint agent types, thus introducing graded agent types: e.g., for any two rules $r_1 : r_d^{\mathrm{OBL}}[c_i = p]$ and $r_2 : r_d^{\mathrm{DES}}[c_j = \sim p]$ we may reframe the type “social” of Table 1 stating that an n-social agent is such that

$$+\partial_{\mathrm{OBL}} p / -\partial_{\mathrm{DES}} {\sim} p \text{ iff } i \leq n$$

Thus the idea of agent type can also be generalized taking into account $\otimes$ constructs.

It is possible to integrate the above classifications by referring to the notion of conversion [9]. Conversions do not have a direct relation with conflict resolution because they simply affect the condition of applicability of rules. However, they contribute to define the cognitive profile of agents because they allow to obtain conclusions modalized by a certain X through the application of rules which are not modalized by X. According to this view, for example, we may have agent types for which, given $p \Rightarrow_{\mathrm{OBL}} q$ and $+\partial_{\mathrm{INT}} p$ we can obtain $+\partial_{\mathrm{INT}} q$. Of course, this is possible only if we assume a kind of norm regimentation, by which we impose that all agents intend what is prescribed by deontic rules. This conversion, in particular, seems appropriate to characterize some kinds of social agent. Other conversions, which, on the contrary, should hold for all realistic

[6] Table 1 covers only some agent types as it focuses on attacks that involve only two rules.

agents are, for example, those that permit to obtain $+\partial_X q$, $X \in \{\mathrm{DES}, \mathrm{INT}, \mathrm{OBL}\}$, from $p \Rightarrow_{\mathrm{BEL}} q$ and $+\partial_X p$ [9]. Table 2 shows the conversions and specifies the agent types with respect to which each conversion seems to be appropriate. We assume to work at least with realistic agents. Since conversions are used only indirectly for conflict resolution but are conceptually decisive for characterizing agents, they provide criteria to specify new agent types. Not all conversion types make sense and so we consider only 9 of 12 cases. At which phase do agent types intervene in the treatment of conflicts? We

$c(\mathrm{BEL}, \mathrm{OBL})$	realistic	$c(\mathrm{BEL}, \mathrm{INT})$	realistic	$c(\mathrm{BEL}, \mathrm{DES})$	realistic
$c(\mathrm{OBL}, \mathrm{DES})$	c-social	$c(\mathrm{OBL}, \mathrm{INT})$	c-strongly-social	$c(\mathrm{DES}, \mathrm{OBL})$	c-deviant
$c(\mathrm{INT}, \mathrm{DES})$	c-stable	$c(\mathrm{DES}, \mathrm{INT})$	c-selfish	$c(\mathrm{INT}, \mathrm{OBL})$	c-strongly-deviant

Table 2. Conversions

argue that classic agent types, but also violation constraints and conversions, play their role mainly in the goal generation phase, because all these features mainly contribute to characterize the motivational profile of the agent.

Example 6 (Running example; continued). Suppose the agent be strongly-social and c-strongly-social, namely, that obligations override intentions and that we accept conversion $c(\mathrm{OBL}, \mathrm{INT})$. So, we obtain the following additional goals:

$$\{\mathrm{OBL}\mathit{PayY660}\$, \mathrm{INT}\mathit{PayY660}\$\}$$

Since r_6 is now stronger than r_5, we obtain OBL*PayY660*\$, while the second goal is derived via r_6 and conversion $c(\mathrm{OBL}, \mathrm{INT})$. This second means that we dropped the previous conclusion that the agent intended the opposite.

Assume now that the theory is also 0-X-constrained, for $X \in \{\mathrm{INT}, \mathrm{OBL}\}$. This means that no violation is permitted. If so, no new intention or obligation can be derived.

Finally, suppose the agent is realistic and 1-stable. Let us add to R_X the rule $r' : a \Rightarrow_{\mathrm{DES}} \neg \mathit{OrderY}$, and to F the fact a. Thus we would obtain $\mathrm{DES}\neg\mathit{OrderY}$, which is in conflict with the conclusion that can be obtained from r_3. Indeed this is the case since an intention overrides a conflicting desire only if the former is a primary intention.

5 Conclusions and Future Work

In this paper we study programming languages for cognitive agents with preferences. We define rule-based agent theories and inference procedures in defeasible logic using a variant of the $\otimes$ operator of [8] in DL for the representation of preferences, and inspired by the BOID architecture [6] separating conflict-detection from -resolution.

We show how to detect and resolve conflicts using preferences. Programming languages for cognitive agents need such fine-grained mechanisms to represent and resolve conflicts among rules for the interaction among mental attitudes – though ways to resolve conflicts must be described abstractly. We define agent types based on preferences. Agent programming languages must describe patterns of ways to deal with conflicts and more generally patterns of agent behavior. Such patterns have been called agent types. Traditional agent types are realistic and committed, other notions introduced before are stable, selfish, social and opportunistic. In this paper we distinguish twelve agent types for attacks and nine for conversions.

Agent programming languages have to distinguish between an abstract language that deals with interaction among mental attitudes, called a deliberation language, and low level procedures to deal with definitions of conflicts based on temporal and causal reasoning, resources, scheduling, and the like. In this paper we assumed that we can use the same deliberation language with preferences as has been used by Dastani and van der Torre [6].

The architecture for cognitive agents can be divided into modules: goal generation module and plan generation module. In this paper we confined ourselves only to the former. Plan generation can be achieved in a similar way. The the inference mechanism (based on defeasible logic) will be used to deduce sequence of actions (plans) that are required to achieve goals. Thus special rules that permit to infer plans must be devised if certain beliefs and goals are given or obtained via the goal generation module. This means that we have to introduce planning rules, where a planning rule has the following format $\phi_1, \ldots, \phi_n : \psi \Rightarrow_p \pi$. The intuition is that a planning rule can be applied if $\phi_1, \ldots, \phi_n$ are derivable from the agent's beliefs, and ψ is derivable from the agent's goals. In addition we have to devise proof conditions, similar in nature to that we have presented for goal generation, which enable us to deal with both complete plans and partial plans, and how to compose them. These issues are left as matter for future research.

References

1. G. Antoniou, D. Billington, G. Governatori, and M.J. Maher. A flexible framework for defeasible logics. In *Proc. AAAI-2000*, Menlo Park, CA, 2000. AAAI/MIT Press.
2. F.M.T. Brazier, B. Dunin Keplicz, N. Jennings, and J. Treur. Desire: Modelling multi-agent systems in a compositional formal framework. *Int. J. Coop. Inf. Syst.*, 6:67–94, 1997.
3. G. Brewka, S. Benferhat, and D. Le Berre. Qualitative choice logic. *Artificial Intelligence*, 157:203–237, 2004.
4. J. Broersen, M. Dastani, J. Hulstijn, and L. van der Torre. Goal generation in the BOID architecture. *Cog. Sc. Quart.*, 2(3-4):428–447, 2002.
5. M. Dastani, F. de Boer, F. Dignum, and J.-J. Meyer. Programming agent deliberation. In *Proc. AAMAS'03*. 2003.
6. M. Dastani and L.W.N. van der Torre. Programming BOID-plan agents: Deliberating about conflicts among defeasible mental attitudes and plans. In *Proc. AAMAS 2004*, New York, 2004. ACM.
7. M. Dastani, B. van Riemsdijk, F. Dignum, and J.-J. Meyer. A programming language for cognitive agents: Goal directed 3APL. In *Proc. ProMAS'03*. 2003.
8. G. Governatori and A. Rotolo. A Gentzen system for reasoning with contrary-to-duty obligations. In A. Jones and J. Horty, editors, *Proc. Δeon'02*, London, May 2002. Imperial College.
9. G. Governatori and A. Rotolo. Defeasible logic: Agency, intention and obligation. In A. Lomuscio and D. Nute, editors, *Proc. Δeon'04*, Berlin, 2004. Springer.