Programming Cognitive 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, email: mehdi@cs.uu.nl
[2] School of ITEE, University of Queensland, email: guido@itee.uq.edu.au
[3] CIRSFID, University of Bologna, email: rotolo@cirsfid.unibo.it
[4] CWI, Amsterdam, and Delft University of Technology, email: torre@cwi.nl

Abstract. Defeasible Logic is extended to programming languages for cognitive agents with preferences and actions for planning. We define rule-based agent theories that contain preferences and actions, together with inference procedures. We discuss patterns of agent types in this setting. Finally, we illustrate the language by an example of an agent reasoning about web-services.

1 Introduction

This paper combines two perspectives: (a) a cognitive account of agents that specifies their mental attitudes; (b) modelling agents' behaviour by means of normative concepts. For the first approach, our background is the belief-desire-intention (BDI) architecture, where mental attitudes are taken as primitives to give rise to a set of Intentional Agent Systems [16,3]. This view is interesting especially when the behaviour of agents is the outcome of a rational balance among their (possibly conflicting) mental states. The normative aspect is rather based on the assumption that normative concepts play a role to characterize the idea of social co-ordination of autonomous agents [15]. The combination of these perspectives leads to an account of agents' deliberation and behaviour in terms of the interplay between mental attitudes and normative (external) factors such as obligations.

Given this background, several rule-based approaches are available for programming cognitive agents [5,9,4]. In this paper we extend the Defeasible Logic (DL) approach. As is well-known, DL is based on a logic programming-like language and it 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. In addition, several efficient implementations have been developed [14,2]. Here we propose a non-monotonic logic of agency, based on the framework of [1], which extends the preliminary work we presented in [7]. Indeed, DL is one of the most expressive languages that allows for the definition of large sets of patterns called agent types. Moreover, it is flexible to incorporate ideas from other languages, such as extension generation and selection from BOID [5], or deliberation languages from 3APL [9,6].

However, as we argued in [7], it has two limits. 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

G. Sutcliffe and A. Voronkov (eds)
LPAR 2005.
LNCS 3835, pp. 621–635, 2005.

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 [9] and in recent incarnations of the BOID architecture [8].

Some issues on agent programming languages should be addressed: how to detect and resolve conflicts that include such preferences, and which kind of agent types can be introduced to deal with preferences. We contribute to cognitive agent programming languages by addressing the following question: How to use DL extended with actions and graded preferences? This question breaks down in the following sub-questions: (a) How to introduce preferences and actions for planning in DL? (b) How to detect and resolve conflicts using preferences and actions? (c) How to define agent types based on preferences and actions?

We provided in [7] some first intuitions on the question which kind of preferences can be introduced in DL. In particular, we reconsidered the introduction of the $\otimes$ operator of [11] in DL, given its advantages over other comparative notions. First, we argued that it can be integrated with a rule based formalism (see also [10]). Second, it has been applied to complicated problems in deontic logic [11]. Third, it allows to clearly distinguish between conflicts and violations [10,11]. In fact, though these notions may conflate, conflicts and violations have in general to be kept separate. Suppose you have an agent doing B while an obligation states OBL$\neg B$. Since the logic for OBL is usually not reflexive[5], the scenario does not lead necessarily to a logical conflict but a violation: conflict-resolution strategies may require that OBL$\neg B$ is not overridden. This paper provides a further step as it provides a more extensive treatment of conflict-detection and -resolution strategies. In addition, it discusses a more comprehensive classification of agent types.

A second substantial step of this work is that it shows how DL can embed a machinery for dealing with planning agents. In this regard, to attack the questions with respect to complex actions in BOID, [8] separate conflict-detection from -resolution. They ask the question whether two plans are conflicting or not, and they ask the question how to resolve conflicts between plans. Analogously, we use the distinction between conflict-detection and -resolution for the $\otimes$ constructions too. This asks for another way to deal with the notion of conflict.

We will distinguish between goal (desires, intentions, obligations) generation and plan generation. The goal generation generates goals based on existing beliefs, desires, intentions and obligations, and the plan generation generates sequences of actions based on these goals. As for the first aspect, rules will allow the derivation of 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 the knowledge an agent has about the world: beliefs may contribute to derive goals (desires, intentions, and obligations), but they are not in themselves motivations

[5] As is well-known, in a non-reflexive modal logic A does not follow from XA, where X is a modal operator.

for action. As for the second aspect, the inference mechanism will be used to deduce sequences of actions (plans) to achieve goals.

The layout of this paper is as follows. In Section 2 we introduce agents with preferences and actions in DL, and in Section 3 we show how to infer goal conclusions from rules with preferences. In Section 4 we discuss how to integrate the previous framework to reason about plans in DL. Finally, in Section 5 we extensively 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 modelling agent's deliberation and behaviour.

Accordingly the formal language contains modal literals, preferences, and actions, and is defined as follows:

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

- *The goal language L_{goal} is the smallest set containing modal literals Xl and $\neg Xl$ when $l \in L$ is a literal and $X \in M$ is a modal operator, and $\otimes$-expressions $l_1 \otimes \ldots \otimes l_n$ when $l_1, \ldots, l_n \subseteq L$ are $n \geq 1$ literals.*
- *The plan language L_{plan} is the smallest set containing Act (basic action plan), $l?$ for all literals l (test action plan), $Achieve(\psi)$ for $\psi \in L$ (abstract action plan), ε (empty plan), and if $\pi, \pi' \in L_{plan}$, then $\pi; \pi'$ (first do π then π'), $\pi | \pi'$ (choose either π or π'), $\pi \parallel \pi'$ (do π and π' simultaneously), π^* (repeat doing π) are in L_{plan} (composite plans). As usual we assume $\forall \pi \in L_{plan} : \varepsilon; \pi = \pi; \varepsilon = \pi$.*

An abstract action plan, $Achieve(\psi)$, can be considered as the representation of a plan which will achieve the goal ψ when it is executed. Moreover, we call a plan π a partial plan if an abstract action occurs in π. A plan in which no abstract action occurs is called a total plan. When the difference is irrelevant, we use the term *plan* to indicate either a partial or a total plan.

For $X \in \{\mathrm{BEL}, \mathrm{INT}, \mathrm{DES}, \mathrm{OBL}\}$, we have that $\phi_1, \ldots, \phi_n \rightarrow_X \psi$ is a *strict rule* such that whenever the premises $\phi_1, \ldots, \phi_n$ are indisputable so is the conclusion ψ. $\phi_1, \ldots, \phi_n \Rightarrow_{X \cup \{p\}} \psi$ is a *defeasible rule* that can be defeated by contrary evidence. A rule $\phi_1, \ldots, \phi_n \rightsquigarrow_X \psi$ is a *defeater* that is used to defeat some defeasible rules by supporting 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 $\rightsquigarrow$ for a defeater), and its* consequent *$C(r)$ (or* head*). In addition the arrow is labelled either with a modal operator $X \in \{\mathrm{BEL}, \mathrm{DES}, \mathrm{INT}, \mathrm{OBL}\}$ or p (only for defeasible* *rules*[6]*). If the arrow is labelled with* BEL *the rule is for belief, and similarly for the other modal operators; if it is labelled with p, then the rule is a planning rule.*

- *A goal rule is a rule r, where $A(r)$ is a set of literals or modal literals, and $C(r)$ is a literal for strict rules, and an $\otimes$-expression for defeasible rules and defeaters.*
- *A planning rule is a defeasible rule of the form $\phi_1, \ldots, \phi_n : \psi \Rightarrow_p \pi$ where $\pi \in L_{plan}$, and $\phi_1, \ldots, \phi_n, \psi \in L_{goal}$ are literals or modal literals.*
- *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 $\otimes_{i=1}^n c_i$.*

The purpose of goal generation is to derive modalised literals (with the exception of rules for beliefs, which are meant to constitute the reasoning core of the system). For example, the application of $p \Rightarrow_{\mathrm{INT}} q$ permits to infer INTq.

Accordingly, modalities will not occur in the consequents of rules to keep the system manageable. We also impose that action symbols may occur only in planning rules.

Definition 3 (Defeasible agent theory). *A defeasible agent theory is a structure $D = (F, R^{\mathrm{BEL}}, R^{\mathrm{DES}}, R^{\mathrm{INT}}, R^{\mathrm{OBL}}, R^p, >)$ 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, R^p is a set of planning rules, 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 or conflicting conclusions, we can establish a preference over and within complex conclusions by using the operator $\otimes$.

In fact, the intuitive reading of a sequence like $a \otimes b \otimes c$ is that a is preferred, but if $\neg a$ is the case, then b is preferred; if $\neg b$ is the case, given $\neg a$, then the third choice is c.

Definition 4 (Preference operator). *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).*

[6] We assume that planning rules are only defeasible. Since their intuitive role is to infer the plans that allow the achievement of the goals of their antecedents, it may seem odd that planning rules may be defeaters, e.g., rules that only block inferences. Indeed, it could be argued that a defeater $\phi_1, \ldots, \phi_n : \psi \rightsquigarrow \pi$ intuitively can be just used to prevent the conclusion of a plan π' that is is incoherent with regard to another plan which would lead to ψ. But the conceptual plausibility of this reading strongly depends on the precise account we provide for the notion of coherence of plans. Since we do not not commit ourselves to any specific interpretation of this notion, we prefer not to consider this case here. We also assumed that planning rules cannot be strict. Suppose to have two planning rules with the same antecedent a but with consequents α and β. Intuitively, we could expect that these rules generate a new rule with the antecedent a and with the consequent $\alpha|\beta$. However, we will not discuss these cases to keep the system manageable.

The general idea of degree of preferences and $\otimes$ formulas are interpreted as preference formulas like in [11] and are here extended to cover all motivational components (but $\otimes$-expressions will not occur in planning rules). Let us see some examples to see the intuitive meaning of such extension:

For beliefs, rule $\neg \mathit{SunShining} \Rightarrow_{\mathrm{BEL}} \mathit{Raining} \otimes \mathit{Snowing}$ says 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 a purchase order, then she will be obliged to pay, but, in the event this is not done, she will have to pay interest.

According to the reading proposed for $\otimes$, suppose we have a rule for obligation such as $a \Rightarrow_{\mathrm{OBL}} b \otimes c$: if a is given, it says that b is obligatory; but, if $\neg b$, then c is obligatory. A similar intuition applies to the other types of rules.

Example 1. (Running example) Suppose an agent desires an application server. She can buy two products from X or Y. 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 does not intend to pay to Y 660\$. Agent's financial resources amount to 700\$, which are available in 4 days. We also know that the agent is a Linux user, and has a credit card and a bank account. With $X \in \{\mathrm{BEL}, \mathrm{DES}, \mathrm{INT}, \mathrm{OBL}\}$, this piece of theory is used to derive goals.

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

Making an order requires to send the order. However, the plan theory does not specify how to achieve this goal with X. On the other hand, sending an order to Y requires to provide agent's data and send them. Y allows to pay either by bank transfer, which requires to provide a digital signature, bank data of Y and to specify the amount of 660\$, or by credit card, which requires to send credit card data and specify the amount. It is not possible to pay by a bank transfer *and* by credit card. The following piece of theory is considered for generating agent's plans (bold symbols denote actions):

$$\begin{aligned}
R^p = \{ & r_8 : \top : \mathit{OrderX} \Rightarrow_p \mathit{Achieve}(\mathit{SendOrderX}),\ r_9 : \top : \mathit{OrderY} \Rightarrow_p \mathit{Achieve}(\mathit{SendOrderY}) \\
& r_{10} : \top : \mathit{SendOrderY} \Rightarrow_p \mathbf{ProvData}; \mathbf{SendDataToY} \\
& r_{11} : \mathit{BAccount} : \mathit{PayY660\$} \Rightarrow_p \mathit{Achieve}(\mathit{TransferY660\$}) \parallel \neg \mathit{Achieve}(\mathit{Pay660\$CCard}) \\
& r_{12} : \mathit{CCard} : \mathit{PayY660\$} \Rightarrow_p \neg \mathit{Achieve}(\mathit{TransferY660\$}) \parallel \mathit{Achieve}(\mathit{Pay660\$CCard}) \\
& r_{13} : \top : \mathit{TransferY660\$} \Rightarrow_p \mathbf{DigitalSign}; \mathbf{ProvBankDataY}; \mathbf{Spec660}\$ \\
& r_{14} : \top : \mathit{Pay660\$CCard} \Rightarrow_p \mathbf{SendToYCreditCardData} \parallel \mathbf{Spec660}\$ \} \\
>= \{ & r_{11} > r_{12} \}
\end{aligned}$$

3 Goal generation: inference with preferences

Definition 5 (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.*

The meaning of the proof tags $+\Delta$, $-\Delta$, $+\partial$ and $-\partial$ is as follows: $+\Delta_X q$ 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, $+\partial_X q$ that q is defeasibly provable in D and $-\partial_X q$ that q is not defeasibly provable.

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 6. *Let $\# \in \{\Delta, \partial\}$, 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 modalised with the modal operator we want to prove. Thus, for example, given the rule $p,q \rightarrow_{\mathrm{BEL}} s$, we can derive $+\Delta_Y s$ if we have $+\Delta_Y p$ and $+\Delta_Y q$.

Conditions for ∂_X are more complicated since we have to consider $\otimes$-expressions. 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 [12]. In this paper 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 modalised 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 also Section 5).

Definition 7. *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* $-\partial_{\mathrm{BEL}} a$ *or* $-\partial_X a$ *for some* $a \in A(r)$.

Example 2. Rule $a, \mathrm{INT}b \Rightarrow_{\mathrm{BEL}} c$ is applicable if we can prove $+\partial_{\mathrm{BEL}} a$ and $+\partial_{\mathrm{INT}} b$.

Remark 1. The notion of conversion is not strange. In many formalisms we can convert from one type of conclusion into a different one. Take for example the right weakening rule of non-monotonic consequence relations, where it is possible to combine non-monotonic consequence with classical consequences: $B \vdash C$ and $A \mid\!\sim B$ imply $A \mid\!\sim C$ [13]. Here, conversions will simply allow to obtain conclusions modalised by a certain X through the application of rules which are not modalised by X.

Example 3. If we have a type of agent that allows a deontic rule to be converted into a rule for intention, $c(\mathrm{OBL},\mathrm{INT})$, then the definition of applicable in the condition for $\pm\partial_{\mathrm{INT}}$ is as follows: a rule $r \in R_{sd}[c_i = q]$ is applicable iff (1) $r \in R^{\mathrm{INT}}$ 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 $\forall a \in A(r)$, $+\partial_{\mathrm{INT}} a \in P(1..n)$. In this second case, for example, given the rule $p,q \Rightarrow_{\mathrm{OBL}} s$, we can derive $+\partial_{\mathrm{INT}} s$ if we have $+\partial_{\mathrm{INT}} p$ and $+\partial_{\mathrm{INT}} q$.

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 $\forall i' < i$, $-\partial_{\mathrm{BEL}} c_{i'} \in P(1..n)$; 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
$\forall k' < k$, $-\partial_{\mathrm{BEL}} c_{k'} \in P(1..n)$ 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 $+\partial_{\mathrm{BEL}} c_{i'} \in P(1..n)$, 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 $+\partial_{\mathrm{BEL}} c_{k'} \in P(1..n)$ or $t \not> s$

For defeasible rules we deal with $\otimes$ formulas. 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 a theory 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.

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 directly generate goals.

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

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

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 Plan generation

A planning rule $\phi_1, \ldots, \phi_n : \psi \Rightarrow_p \pi$ may be intuitively read as a rule that allows for the derivation of a plan π that permits to achieve a single goal ψ, given the beliefs $\phi_1, \ldots, \phi_n$. In other words, such a rule can be applied if $\phi_1, \ldots, \phi_n$ are believed, i.e. if they are derivable from the agent's beliefs, and ψ should be achieved, i.e. if ψ is derivable from the agent's goals. This implies that we will have various conclusions for goal formulae and thus the following tagged literals: $+\Delta_G p$, $-\Delta_G p$, $+\partial_G p$, $-\partial_G p$ where $G \in \{\text{DES}, \text{INT}, \text{OBL}\}$. Similar to the definition of derivations of tagged literals, we define the notion of provability of plans. In the following, we use $A^B(r)$ to denote the belief conditions of the planning rule r, and $A^G(r)$ to denote its goal condition. For example, for the planning rule $r = \phi : \psi \Rightarrow_p \pi$, we have $\phi \in A^B(r)$ and $A^G(r) = \psi$. A plan π is derivable if no plan π' is derivable which is incoherent with π. The notion of coherence of plans is the counterpart of the notion of consistency of logical formulae which is used for the provability of literals. The notion of coherence can be defined, for example, in terms of resource conflicts or possibility of plan execution. We will not enter here into a detailed discussion of this issue. However, we can formulate a very minimal condition for compatible plans in terms of the belief and goal conditions of rules that generate them. In particular, two plans are compatible iff the belief and goal conditions of the rules applied to their derivations are consistent. This fact is already embedded in our framework because the goal generation phase described in this paper provides criteria for deriving consistent goals. The only exceptions are when facts (not derived goals) are inconsistent or, we will see in Section 5, when the agent type adopted permits to obtain, for example, that $+\partial_{\text{OBL}} a$ and $+\partial_{\text{INT}} \neg a$. In these cases, the superiority relation that may apply specifically to planning rules can be decisive. In fact, given the possibility to obtain $+\partial_{\text{OBL}} a$ and $+\partial_{\text{INT}} \neg a$, two planning rules $\top : a \Rightarrow_p \pi$ and $\top : \neg a \Rightarrow_p \pi'$ turn out to be both applicable. However, although for certain agent types $+\partial_{\text{OBL}} a$ and $+\partial_{\text{INT}} \neg a$ do not correspond to a conflict (OBLa and INT$\neg a$ are not necessarily in contradiction), it may be argued that the plans leading to achieve a and $\neg a$ are incoherent (intuitively incompatible). Notice also that the plan language introduced in Section 2 does not admit the negation of action symbols. So, in theory, logical inconsistency is not relevant as regards the derivation of plans (the consequents of planning rules). However, we may also have partial plans that include special abstract actions to achieve goals. In this case, logical consistency of derived plans and the corresponding conflict resolution may play a role as in the phase of goal generation.

Let us see first the basic proof conditions for the generation of total plans, i.e., plans in which no abstract actions occur.

$+\Pi$: If $P(i+1) = +\Pi\pi$ then
(1) $\exists r \in R^p[\pi]$ such that
(1.1) $\phi_1, \ldots, \phi_n \in A^B(r)$ and $A^G(r) = \psi$, and
(1.2) $\forall k, 1 \leq k \leq n$, $+\partial_{\text{BEL}} \phi_k \in P(1..i)$ and $+\partial_G \psi \in P(1..i)$, and
(2) $\forall s \in R^p[\pi']$ such that incoherent(π, π') either
(2.1) $\exists \phi' \in A^B(s) : -\partial_{\text{BEL}} \phi' \in P(1..i)$ or
(2.2) $\psi' \in A^G(s) : -\partial_G \psi' \in P(1..i)$ or
(2.3) $\exists t \in R^p[\pi]$ such that $t > s$ and $\forall \phi'' \in A^B(t) : +\partial_{\text{BEL}} \phi'' \in P(1..i)$ and $\psi'' \in A^G(t) : +\partial_G \psi'' \in P(1..i)$.

Thus, a total plan is defeasibly derivable if the conditions (1) and (2) hold. Condition (1) states that a total plan π is defeasibly derivable at derivation step $P(i+1)$ if there exists a planning rule with π as its consequent such that its belief and goal conditions are defeasibly provable at derivations $P(1..i)$. Condition (2) states that if there exists a planning rule s such that its consequent is the total plan π' which is incoherent with plan π, then either the belief and goal conditions of rule s are not defeasibly derivable or there exists a preferred planning rule t with plan π as its consequent for which its beliefs and goals are defeasibly derivable. Note that we assume that a planning rule is applicable if its belief and goal conditions are defeasibly provable. We may also consider the case where the belief and goal conditions are definitely provable.

Analogously, we define the non-provability of total plans $-\mathit{\Pi}\pi$ as follows:

$-\mathit{\Pi}$: If $P(i+1) = -\mathit{\Pi}\pi$ then
(1) $\forall r \in R^p[\pi]$ either
(1.1) $\exists \phi \in A^B(r)$ and $-\partial_{\text{BEL}}\phi \in P(1..i)$ or
(1.2) $A^G(r) = \psi$ and $-\partial_G\psi \in P(1..i)$, or
(2) $\exists s \in R^p[\pi']$ such that incoherent(π ,π') and
(2.1) $\forall \phi' \in A^B(s) : +\partial_{\text{BEL}}\phi' \in P(1..i)$ and
(2.2) $\psi' \in A^G(s) : +\partial_G\psi' \in P(1..i)$ and
(2.3) $\forall t \in R^p[\pi]$ either $t \not> s$ or $\exists \phi'' \in A^B(t) : -\partial_{\text{BEL}}\phi'' \in P(1..i)$ or
$\psi'' \in A^G(t) : -\partial_G\psi'' \in P(1..i)$.

Thus, a total plan is not defeasibly provable if one of the conditions (1) or (2) holds. Condition (1) states that a total plan π is not defeasibly derivable at derivation step $P(i+1)$ if the belief or goal conditions of all planning rules with π as its consequent are not defeasibly provable at derivations $P(1..i)$. Condition (2) states that if there exists a planning rule (s) such that its consequent is the total plan π' which is incoherent with plan π, then its belief and goal conditions are defeasibly derivable and, moreover, for all more preferred planning rules t with the total plan π as its consequent it is the case that their beliefs or goals are not defeasibly derivable.

This definition of plan provability should be modified to allow the derivation of plans that are obtained from the application of planning rules to refine an existing partial plan. In order to define this notion of plan provability, we first assume the function $occurs(\psi,\pi)$, which returns true if the abstract action $Achieve(\psi)$ occurs in the partial plan π, and the function $sub(\psi,\pi',\pi'')$, which returns a plan by substituting the abstract action $Achieve(\psi)$ in π' with plan π''. For example, consider the partial plan $\pi = \alpha;Achieve(\psi);\beta$. Then, $occur(\psi,\pi) = true$ and $sub(\psi,\pi,\gamma|\delta) = \alpha;(\gamma|\delta);\beta$. The definition of defeasible provability of plans which involve abstract actions, indicated by $+\mathit{\Omega}\pi$, can be defined as follows:

$+\mathit{\Omega}$: If $P(i+1) = +\mathit{\Omega}\pi$ then either
(1) $+\mathit{\Pi}\pi \in P(1..i)$, or
(2) $+\mathit{\Omega}\pi' \in P(1..i)$ such that
(2.1) $\exists r \in R^p[\pi'']$ and
(2.2) $\phi_1,\ldots,\phi_n \in A^B(r)$ and $A^G(r) = \psi$, and
(2.3) $\forall k, 1 \le k \le n$, $+\partial_{\text{BEL}}\phi_k \in P(1..i)$ and $+\partial_G\psi \in P(1..i)$ and
(2.4) $occurs(\psi,\pi')$ and $sub(\psi,\pi',\pi'') = \pi$.

Thus, a plan is defeasibly provable if one of the conditions (1) or (2) holds. Condition (1) states that a plan is provable if it is provable directly by applying planning rules.

Condition (2) states that a plan is derivable if there exists a partial plan which can be refined by applying a rule.

Analogously, for plans that involve abstract actions we define the non-provability of plans $-\Omega\pi$ as follows:

$-\Omega$: If $P(i+1) = -\Omega\pi$ then
(1) $-\Pi\pi \in P(1..i)$, and
(2) $+\Omega\pi' \in P(1..i)$ such that
  (2.1) $occurs(\psi,\pi')$ and $sub(\psi,\pi',\pi'') = \pi$ and
  (2.2) $\forall r \in R^p[\pi'']$:
    (2.2.1) $\phi_1,\dots,\phi_n \in A^B(r)$ and $\exists k$, $1 \leq k \leq n$, $-\partial_{\mathrm{BEL}}\phi_k \in P(1..i)$ or
    (2.2.2) $A^G(r) = \psi$ and $-\partial_G\psi \in P(1..i)$.

Thus, a plan is not defeasibly provable if the conditions (1) or (2) hold. Condition (1) states that a plan is not provable if it is not directly provable and condition (2) states that the plan is not provable through applications of planning rules to partial plans.

Example 5 (Running example; continued). Given the conclusions derived in Section 3, let us consider the only positive goal, namely $+\partial_{\mathrm{INT}}OrderY$. However, assume, as we will do in Example 6, to have also $+\partial_{\mathrm{OBL}}PayY660\$$ and $+\partial_{\mathrm{INT}}PayY660\$$. These goals make planning rules r_9, r_{11} and r_{12} applicable, whereas $\mathrm{INT}\neg OrderX$ makes r_8 non-applicable. r_9 includes an abstract plan to be specified. This is possible via r_{10}. On the other hand, the agent has to pay 660$ to Y, but has to choose between two incompatible plans: paying using the credit card of by bank transfer. Here r_{11} and r_{12} provide each simultaneous partial plans that dictate to make a bank transfer and not paying by credit card or the opposite. Since $r_{11} > r_{12}$, the agent prefers the latter option. The derived total plans are then

$$\{\mathbf{ProvData};\mathbf{SendDataToY},\ \mathbf{DigitalSign};\mathbf{ProvBankDataY};\mathbf{Spec660}\$\}$$

Finer criteria for dealing with provability in plans may be introduced when finer criteria are used in the goal generation. If the agent is realistic and 1-stable, as we will see in Section 5, then $-\partial_{\mathrm{INT}}OrderY$; thus we cannot derive, too, any plan.

5 Conflict resolution and agent types

At which phase do agent types intervene in the treatment of conflicts, and how can they be generalised to incorporate $\otimes$ formulas? Classically, agent types are characterised by stating conflict resolution types in terms of orders of overruling between rules [5,12]. For example, an agent is *realistic* when rules for beliefs override all other components; she is *social* when obligations are stronger than the other motivational components with the exception of beliefs. Agent types can be characterised 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 $r > r'$. Then let us see the agent types that can be identified in the framework we have defined so far. Table 1 shows all possible cases and, for each kind of rule, indicates all attacks on it. It should be read as follows. Each of the three main columns identifies a possible kind of conflict between two types X,Y of applicable rules that would permit to infer the literals p and $\sim p$ labelled by X and Y respectively. The first two sub-columns in each main column indicate whether both literals are derived (i.e., there is no real conflict, which is indeed a logical possibility since we are dealing with modalities which do not enjoy reflexivity), or whether we have conflict where one rule prevails over the other, or where the two rules defeat each other. Finally, the third sub-column defines the agent type for which each conflict-detection and -resolution policy is appropriate. Since we have to consider three kinds of rules for generating goals, we have to analyse twelve combinations. (To save space, in Table 1 “s-” is an abbreviation for “strongly-”; “indep.” abbreviates “independent”.)

$r_d^{\mathrm{OBL}}[c_i = p]/\, r_d^{\mathrm{INT}}[c_j = \sim p]$			$r_d^{\mathrm{OBL}}[c_i = p]/\, r_d^{\mathrm{DES}}[c_j = \sim p]$			$r_d^{\mathrm{INT}}[c_i = p]/\, r_d^{\mathrm{DES}}[c_j = \sim p]$		
$+\partial_{\mathrm{OBL}} p$	$+\partial_{\mathrm{INT}} \sim p$	s-indep.	$+\partial_{\mathrm{OBL}} p$	$+\partial_{\mathrm{DES}} \sim p$	indep.	$+\partial_{\mathrm{INT}} p$	$+\partial_{\mathrm{DES}} \sim p$	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: Basic Attacks

Independent and strongly-independent agents are free respectively to adopt desires and intentions in conflict with obligations. As expected, for social and strongly-social agents obligations override desires and intention. 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 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.

Table 1 does not cover all possible types of agent. In fact, the table focuses on possible attacks that involve only two rules; in addition we will assume that belief rules are always stronger than intentions, desires and obligations. This is motivated by the intuition that belief rules describe specification of the environment where the agent is situated. Table 2 completes the scenario and provides all possible combinations when we deal with three rules, in particular, we consider all possible relationships between obligation rules on one side and intention and desire rules on the other side. For example we consider agent types where an obligation rule can be defeated by an intention rule and, at the same time, it can defeat a desire rule (social-strongly social). This allows for the specification of new agent types based on the basic types defined in Table 1.

$r_d^{\mathrm{OBL}}[c_i = p]/r_d^{\mathrm{INT}}[c_j = \sim p]/r_d^{\mathrm{DES}}[c_k = \sim p]$			
$+\partial_{\mathrm{OBL}} p$	$+\partial_{\mathrm{INT}} \sim p$	$+\partial_{\mathrm{DES}} \sim p$	hyper-independent
$+\partial_{\mathrm{OBL}} p$	$+\partial_{\mathrm{INT}} \sim p$	$-\partial_{\mathrm{DES}} \sim p$	social-strongly-independent
$+\partial_{\mathrm{OBL}} p$	$-\partial_{\mathrm{INT}} \sim p$	$+\partial_{\mathrm{DES}} \sim p$	social-independent
$+\partial_{\mathrm{OBL}} p$	$-\partial_{\mathrm{INT}} \sim p$	$-\partial_{\mathrm{DES}} \sim p$	hyper-social
$-\partial_{\mathrm{OBL}} p$	$+\partial_{\mathrm{INT}} \sim p$	$+\partial_{\mathrm{DES}} \sim p$	hyper-deviant
$-\partial_{\mathrm{OBL}} p$	$+\partial_{\mathrm{INT}} \sim p$	$-\partial_{\mathrm{DES}} \sim p$	social-strongly-deviant
$-\partial_{\mathrm{OBL}} p$	$-\partial_{\mathrm{INT}} \sim p$	$+\partial_{\mathrm{DES}} \sim p$	social-deviant
$-\partial_{\mathrm{OBL}} p$	$-\partial_{\mathrm{INT}} \sim p$	$-\partial_{\mathrm{DES}} \sim p$	hyper-pragmatic

Table 2. Agent Types: Other Attacks

However, this taxonomy can be enriched thanks to the role that may be 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 m violations for each rule with g in the head:

Definition 8 (Violation constraint on goals). *Let m and X be an integer and a type of rule, respectively. A theory D will be m-X-constrained iff, given the definition of $+\partial$, for all literals q, $+\partial_X q$ iff (1) $i' \leq m$; and (2) if $1 \leq j' \leq j$ and $s \in R^X$, then $j' \leq m$; and (3) $k' \leq m$. 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 m-social agent is such that

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

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

It is possible to integrate the above classifications by referring to the notion of conversion [12]. Conversions do not have a direct relation with conflict resolution because they simply affect the condition of applicability of rules. However, they indeed contribute to define the cognitive profile of agents because they allow to obtain conclusions modalised by a certain X through the application of rules which are not modalised 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 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 $+\partial_X p$ [12]. Table 3 shows the conversions and specify 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 characterising agents, they provide criteria to specify new agent types. Not all conversion types make sense and so we consider only 9 cases out of 12 possible combinations.

$c(\mathrm{BEL}, \mathrm{OBL})$	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 3. Conversions

At which phase do agent types intervene in the treatment of conflicts? Classic agent types, 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. Notice, however, that we could also introduce $\otimes$ in plans. With plans, in fact, we would need as well a finer distinction than just assuming that either two plans conflict or they do not; for example, in [8] no finer distinction was made. In particular, $\otimes$ in planning rules could express non-deterministic effects of actions. However, we prefer here not to do this, to keep the system manageable. This does not mean that we cannot introduce finer criteria for dealing with provability in plans, but this can be simply made just referring to derivation of the goals that occur, as results, in the planning rules. As we have seen, a planning rule $\phi_1,\ldots,\phi_n : \psi \Rightarrow_p \pi$ permits to infer plan π, a plan that is meant to produce the goal ψ given beliefs $\phi_1,\ldots,\phi_n$. Plan π is conceptually the condition for obtaining ψ. Thus, Definition 8 will allow the agent to obtain π only if ψ or $\phi_1,\ldots,\phi_n$ do not require more than m violations for each X rule.

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 $\mathrm{OBL}\mathit{PayY660}\$$, while the second goal is derived via r_6 and conversion $c(\mathrm{OBL},\mathrm{INT})$. This second means that we drop the previous conclusion obtained in Example 4, i.e. that the agent intends 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.

6 Conclusions

In this paper we extend DL with preferences and actions. We show how to detect and resolve conflicts using preferences and actions. Rule based languages follow the tradition of production rules in knowledge based systems and logic programming. The extension of production rules is based on the use of rule based systems in cognitive attitudes in practical reasoning. Indeed, the new issue is the interaction among mental attitudes. Examples are Thomason's BDP, programming languages based on the BOID architecture, 3APL, etc. In general, conditional approaches and preference based approaches have been traditionally defined in terms of each other. For example, “if A then B” has been defined as “A and B is preferred to A without B”, and “A is preferred to B” has been defined as “if A or B, then A”. However, it may be unnatural to define preferences in terms of conditionals, and it is more natural to define them directly. Moreover, special preference-based formalisms may be more efficient, such as CP nets. Finally, the kind of preferences which can be expressed in terms of conditionals is only limited to special kinds. This explains why comparative notions are now a major topic of concern in artificial intelligence and practical reasoning.

Let us summarise some requirements for programming cognitive agents. First, the interaction among mental attitudes needs fine-grained mechanisms to represent and resolve conflicts among rules. Second, the programming language has 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. Third, ways to resolve conflicts must be described abstractly. Fourth, patterns of ways to deal with conflicts and more generally patterns of agent behaviour must be described. Such patterns have been called agent types. Fifth, the interaction between mental attitudes and semantics of MAS communication–as defined e.g. by FIPA–should be realised.

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 [8]. Moreover, we did not address the issue of MAS communication, because the mental attitudes approach to communication has been attacked recently by social commitment approaches; a careful reconsideration of this issue is beyond the scope of this paper [17] and is left 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*. AAAI/MIT Press, 2000.
2. N. Bassiliades, G. Antoniou, and I. Vlahavas. DR-DEVICE: A defeasible logic system for the Semantic Web. In *Proc. PPSWR 2004*. Springer, 2004.
3. M. Bratman, D. Israel, and M. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 1988.
4. 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.*, 1997.
5. J. Broersen, M. Dastani, J. Hulstijn, and L. van der Torre. Goal generation in the BOID architecture. *Cog. Sc. Quart.*, 2002.
6. M. Dastani, F. de Boer, F. Dignum, and J.-J. Meyer. Programming agent deliberation. In *Proc. AAMAS'03*. 2003.
7. M. Dastani, G. Governatori, A. Rotolo, and L. van der Torre. Preferences of agents in defeasible logic. In *Proc. AI05*. Springer, 2005.
8. 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*. ACM, 2004.
9. 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.
10. G. Governatori. Representing business contracts in RuleML. *Int. J. Coop. Inf. Syst.*, 2005.
11. G. Governatori and A. Rotolo. A Gentzen system for reasoning with contrary-to-duty obligations. In *Proc. Δeon'02*. Imperial College, 2002.
12. G. Governatori and A. Rotolo. Defeasible logic: Agency, intention and obligation. In *Proc. Δeon'04*. Springer, 2004.
13. S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 1990.
14. M.J. Maher, A. Rock, G. Antoniou, D. Billignton, and T. Miller. Efficient defeasible reasoning systems. *Int. J. Art. Int. Tools*, 2001.
15. J. Pitt, editor. *Open Agent Societies*. Wiley, Chichester, 2004.
16. A. Rao and M. Georgeff. Modelling rational agents within a BDI-architecture. In *KR'91*. Morgan Kaufmann, 1991.
17. M.P. Singh. A social semantics for agent communication languages. In *Issues in Agent Communication*. Springer, 2000.