Combining Goal Generation and Planning in an Argumentation Framework

Joris Hulstijn
ICS, Utrecht University
jorish@cs.uu.nl

Leendert van der Torre
CWI, Amsterdam
torre@cwi.nl

Abstract

We study conflicts between goals and plans in Dung's abstract argumentation framework. Argumentation theory has traditionally been used to deal with conflicts between defaults and beliefs. Recently Amgoud has proposed to use it for conflicts between plans. Amgoud argues that Dung's argumentation theory has to be adjusted, because conflicts between plans are fundamentally different from conflicts between defaults. We agree with the fundamental difference, but we propose an alternative way to deal with conflicts between plans that stays within Dung's framework. Moreover, we extend Amgoud's argumentation framework for planning with goal generation procedures, that can deal with conflicts between goals. In the proposed framework goals are derived from desires by forward reasoning, and plans are derived from goals and planning rules by backward reasoning.

Introduction

Formal argumentation theory originates from theories of dialogue and natural language, but it has become popular in artificial intelligence as a formal framework for default reasoning. Dung's (1995) abstract framework characterizes many instances of non-monotonic reasoning, such as Reiter's default logic, autoepistemic logic, logic programming, and circumscription, as instances of argumentation. Amgoud's version of Dung's argumentation framework allows reasoning about conflicting plans (Amgoud 2003; Amgoud & Cayrol to appear). The central analogy is the following. A goal that has possible 'trees of realization' or plans to achieve it, can be modeled just like an argument which consists of a claim with the supporting argumentations. The attack relation defined over arguments can serve as a criterium to select compatible plans. However, Amgoud argues that the framework must be adapted, because conflicts between plans are fundamentally different from the kinds of conflicts studied in non-monotonic reasoning, such as defaults.

In this paper, we study the use of an argumentation framework for planning extended with goals generated from desires. The research questions of this paper are as follows:

1. What are the differences between conflicts usually studied in argumentation theory, and conflicts between plans?
2. How can we represent conflicts between plans in Dung's argumentation framework, without adapting the central definitions?
3. How can we extend an argumentation framework that deals with conflicts between plans, such that it also correctly deals with conflicts between goals?

The resulting framework leads to new insights in the nature of conflicts in artificial intelligence, and the applicability of argumentation theories to study conflicts. We do not compare the argumentation framework to other approaches to planning. Although the treatment of planning is very coarse, we believe it may be relevant for theories that use resource logic to describe competition between plans.

The paper is structured as follows. First we study conflicts between plans in Amgoud's argumentation framework. Then we introduce our own representation of plans in Dung's argumentation framework. After that we extend the approach with goal generation.

Argumentation framework for planning

In the logical language, we distinguish among decision variables (A) and non-decision variables (N). Decision variables represent atomic actions, hence the A, that need no further plan to be achieved. This distinction is implicit in Amgoud's framework.

Definition 1 Let A and N be two disjoint sets of decision variables and non-decision variables respectively. Let L be a propositional language built from $A \cup N$. A literal l is a variable or its negation. A rule is an ordered nonempty finite list of literals: $l_1 \wedge l_2 \wedge \ldots \wedge l_{n-1} \rightarrow l_n$. We call $l_1 \wedge l_2 \wedge \ldots \wedge l_{n-1}$ the body of the rule, and l_n the head. If $n = 1$ the body is empty and we write l_n.

A desire plan description consists of a set of desired literals (D), a set of plan rules (P), and a set of knowledge and integrity constraint rules (Σ). Amgoud seems to allow any kind of formulas in Σ, but we restrict it to rules (see also definition 3).

Definition 2 Let A, N and L be as defined in definition 1. A *desire-plan description* is a tuple $\langle D, P, \Sigma \rangle$ with D a set of literals from L, P and Σ sets of rules from L, such that the heads of rules in P are built from a variable in N.

These rules behave as production rules. For explanatory reasons, we restrict the language here to conjunction and a syntactic form of negation. More elaborate logics exists in the literature on logic programming. The exact semantics of the rules depends on the particular argumentation system. The general idea is to use these rules to generate extensions. An extension is a set of argumentations or 'plans', depending on how you look at it, that are mutually compatible. In a way, an extension represents a possible intention for an agent to consider. Ideally, an extension will contain plans to fulfill all desired literals, while no integrity constraints are violated. The selection of which extension to execute will be decided by argumentation theoretic notions.

Amgoud (2003) illustrates desire-plan descriptions with the following example . The examples are only meant to illustrate the definitions; further validation of the approach remains necessary (Amgoud & Cayrol to appear).

Example 1 (Travel) Assume variables $N = \{$*ja, fp, t, vac*$\}$ and $A = \{$*w, fr, hop, dr, ag* $\}$ with the interpretation:

ja	journey to Africa	*w*	work
fp	finish paper	*fr*	friend brings tickets
t	get the tickets	*hop*	go to hospital
vac	be vaccinated	*dr*	go to a doctor
		ag	go to the agency

Consider the following desire-plan description:

$$\begin{array}{lcl} D & = & \{ja, fp\} \\ P & = & \{t \wedge vac \rightarrow ja,\ ag \rightarrow t, fr \rightarrow t,\ hop \rightarrow vac, \\ & & \quad dr \rightarrow vac, w \rightarrow fp\} \\ \Sigma & = & \{w \rightarrow \neg ag,\ w \rightarrow \neg dr\} \end{array}$$

The agent desires to travel and to finish a paper. There are several ways to achieve both desires, but if we add $w \rightarrow \neg hop$ as a third rule to Σ, there is no way to achieve both desires.

Actions are defined as tuples $\langle h, H\rangle$ analogous to a claim for h with support H (Amgoud 2003; Dung 1995). Amgoud considers various notions of conflict, of which we only use the strongest one. Moreover, we do not interpret the arrows in Σ as material implications, but as production rules.

Definition 3 An *action* for $\langle D, P, \Sigma\rangle$ is:

- $\langle h, \emptyset\rangle$ for any $h \in A$, called an atomic action; or
- $\langle h, \{l_1, \ldots, l_n\}\rangle$ for any rule $l_1 \wedge \ldots \wedge l_n \rightarrow h \in P \cup \Sigma$.

We say that:

- The *closure* of a set of rules R over a set of literals V, is defined by $Cl(R, V) = \bigcup_{i=0}^{i=\infty} S^i$ with $S^0 = V$ and $S^{i+1} = S^i \cup \{l \mid l_1 \wedge \ldots \wedge l_n \rightarrow l \in R,\ \{l_1, ..., l_n\} \subseteq S^i\}$.
- Actions $\langle h_1, H_1\rangle$ and $\langle h_2, H_2\rangle$ *conflict* iff $Cl(\Sigma \cup \{h_1, h_2\} \cup H_1 \cup H_2) \vdash \bot$.

From these actions Amgoud constructs so-called realization trees, which serve as a way to represent and reason about plans with their motivating desires. Each node of the tree is an action and each child of an action is one of its subactions.[1]

Definition 4 A *realization tree* for a desire $d \in D$, written as $g(d)$, is a finite tree whose nodes are actions such that

- $\langle d, H\rangle$ is the root of the tree, for some H;
- $\langle h, \{l_1, \ldots, l_n\}\rangle$ has exactly n children $\langle l_1, H_1\rangle$, ..., $\langle l_n, H_n\rangle$;
- The leaves of the tree are atomic actions.

[1]Normally in argumentation frameworks there is an additional constraint that a literal does not occur twice in a path; like Amgoud, we do not consider this optimization in this paper.

The realization trees generated by example 1 are shown in figure 1. In contrast to Amgoud, we visualize realization trees with arrows directed from the leaves to the root.

Example 2 (Travel, continued) There are five realization trees, four for the desire *ja*, and one for the desire *fp*. We call the realization trees for *ja* respectively g_1, g_2, g_3 and g_4, based on respectively $\{ag, hop\}$, $\{ag, dr\}$, $\{fr, dr\}$ and $\{fr, hop\}$, and the realization tree for desire *fp* is called g_5, and is based on $\{w\}$.

Amgoud's approach

Just like arguments, realization trees may conflict. Therefore it makes sense to use concepts from argumentation theory. In general, an argumentation framework is defined as a set of arguments with a binary relation that represents which arguments attack which other arguments (Dung 1995). Here, realization trees play the role of arguments; the attack relation is derived from conflicts between actions (Amgoud 2003).

Definition 5 A *system handling conflicting desires* (SHD) is a tuple $\langle G, Attack\rangle$ such that G is a set of realization trees and $Attack$ is a binary relation over G.

Based on the notion of conflict between actions (definition 3), we specify a particular attack relation. The argumentation notions of *defence, preferred extension* and *basic extension* are defined accordingly. However, Amgoud argues that Dung's notion of defence must be adjusted:

> "The semantics of a realization tree is a complete plan to achieve a desire and our aim is to achieve a maximum of desires. The idea is if a given desire d_1 can be achieved with a plan p_1 then if another plan p_2 for the same desire attacks a plan p_3 of another desire d_2, we will accept p_3 to enable the agent to achieve its two desires." (Amgoud 2003)

In the running example, read $d_1 = ja$, $p_1 = g_4$, $p_2 = g_1$, $d_2 = fp$, and $p_3 = g_5$. We write D-defend for Dung's original definition of defence, and A-defend for Amgoud's definition of defence. Moreover, we also give a variant of Amgoud's notion of defence, which we call A′-defend, which also incorporates the condition that p_1 may not itself attack p_3. We believe that A′-defend is more convincing.

Definition 6 Let $\langle G, Attack\rangle$ be an SHD such that G contains all realization trees of a given desire-plan description $\langle D, P, \Sigma\rangle$. Let $S \subseteq G$ and $g, g_1, g_2 \in G$ be (sets of) realization trees. Now we define

- $\langle g_1, g_2\rangle \in Attack$, i.e., g_1 attacks g_2, iff there exist actions a_1 and a_2 in the nodes of g_1 and g_2 respectively, such that a_1 and a_2 conflict.[2]

[2]Reducing a conflict between realization trees to conflicts between actions implies that we can have two realization trees that do not conflict, although there are three actions which conflict with each other, for example the atomic actions a, b and c with integrity constraint $a \wedge b \rightarrow \neg c$. This problem can be solved by generalizing the notion of conflict between realization trees in the obvious way.

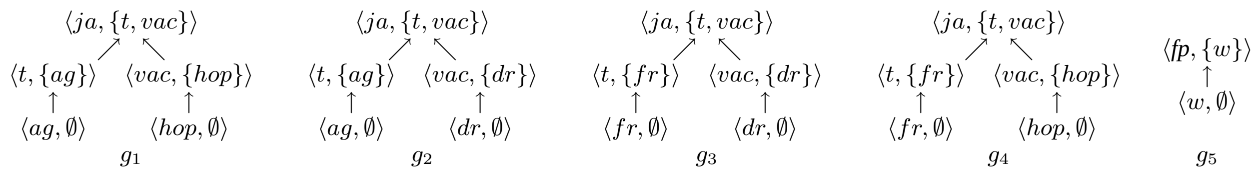

Figure 1: The realization trees for example 1

- S is attack free iff there are no $g_1, g_2 \in S$ such that g_1 attacks g_2.
- S D-defends g iff for all $g_1 \in G$ such that g_1 attacks g, there is an alternative $g_2 \in S$ such that g_2 attacks g_1.
- S A-defends g iff for all $g_1(d_1) \in G$ such that $g_1(d_1)$ attacks g, there is an alternative $g_2(d_2) \in S$ with $d_1 = d_2$.
- S A′-defends g iff for all $g_1(d_1) \in G$ such that $g_1(d_1)$ attacks g, there is an alternative $g_2(d_2) \in S$ with $d_1 = d_2$ and g_2 does not attack g.
- S is a preferred extension iff S is maximal w.r.t. set inclusion among the subsets of G that are attack free and that A-defend all their elements.
- S is a basic extension iff it is a least fixpoint of the function $F(S) = \{g | g \text{ is A-defended by } S\}$.

These argumentation notions are illustrated by the travel example.

Example 3 (Travel, continued) Consider the realization trees in figure 1. The realization tree g_5 attacks both g_1 and g_2, due to the integrity constraint $w \rightarrow \neg ag$, and it attacks both g_2 and g_3 due to the integrity constraint $w \rightarrow \neg dr$, but it does not attack g_4, which is based on *fr* and *hop* instead of *ag* and *dr*.

There are two preferred extensions, $S_1 = \{g_1, g_2, g_3, g_4\}$ for desire ja, and $S_2 = \{g_4, g_5\}$ for desires *ja* and *fp*. Consequently, preferred extensions do not have to be maximal with respect to set inclusion. The basic extension is $\{g_4, g_5\}$, since $F(\emptyset) = \{g_4\}$, $F(\{g_4\}) = \{g_4, g_5\}$, $F(\{g_4, g_5\}) = \{g_4, g_5\}$. This holds also if we replace A-defend by A′-defend.

The two notions of extension still seem to be problematic. As Amgoud shows, for preferred extensions an additional notion is required that prefers extensions that maximize the number of goals achieved. In basic extensions, as Amgoud shows, some of the alternative plans may be conflicting. For example, they may be competing for the same resources. We therefore study a different approach to reason about conflicting plans in argumentation theory.

Our Proposal

Instead of adapting the argumentation theory, we encode conflicting plans in Dung's original framework. Amgoud's reason to alter the definition of defend is that there is a difference between conflicts between the usual kinds of arguments, i.e., defaults or beliefs, and conflicts between plans. We fully agree with this observation. However, instead of changing the notion of defend, we alter the notion of attack. In general, the attack relation is the parameter that covers domain dependent aspects of argumentation. Note that Amgoud's attack relation is symmetrical, which means that the desire plan specification does not have priorities. For Dung's original theory, a symmetrical attack relation produces an argumentation theory which is relatively simple: the (unique) basic extension contains all arguments which are undisputed, whereas preferred extensions – which in this special case always exist – represent maximal sets of arguments for non-conflicting sets of goals. Note moreover, that under Amgoud's definition, all realization trees will self-attack. In our proposal the attack relation does not have to be symmetrical, which can be useful in more complex settings. It no longer is reflexive either.

The basic idea is as follows. We want to achieve a maximal number of desires, and consequently an extension contains a maximal number of realization trees. However, considering different plans for the same desire, may be a waste of resources. We therefore exclude the possibility that an extension contains two realization trees for the same desire. In a sense, this is the opposite of the approach taken by Amgoud. Under her definition a basic extension, for example, may contain several realization trees for the same desire.

Definition 7 Let $\langle G, \mathit{Attack}\rangle$ be an SHD such that G contains all realization trees of a given desire-plan description $\langle D, P, \Sigma\rangle$. Let $S \subseteq G$ and $g, g_1, g_2 \in G$ be (sets of) realization trees.

- $\langle g_1(d_1), g_2(d_2)\rangle \in \mathit{Attack}$, i.e., $g_1(d_1)$ attacks $g_2(d_2)$, iff either
  1. there exist actions a_1 and a_2 in the nodes of g_1 and g_2 respectively, such that a_1 and a_2 conflict, or
  2. $g_1(d_1) \neq g_2(d_2)$ but $d_1 = d_2$.

The definition of attack free remains the same as in definition 6, but in the definition of preferred and basic extension, A-defend is replaced by D-defend.

Our approach can be motivated as follows. We want to model aspects of the deliberation process of an agent. During goal generation, alternative plans for a desire are considered to test its feasibility. Desires may conflict, which is why we use argumentation theory: to reason with multiple extensions. In our view, an extension corresponds to a maximally consistent subset of desires, along with their plans. Thus an extension models a potential goal or an option. Ultimately, an agent must select one option as an intention. Therefore it does not make sense to also consider alternative options within an extension. Keeping alternative options open requires additional deliberation effort (Boella 2002). Moreover, plans may themselves compete for resources and be incompatible for that reason.

The following proposition shows that our formalization of conflicting plans captures the intuition of Amgoud's notion of defence.

Proposition 1 In our argumentation theory, we have that if S D-defends g, then S also A-defends and A$'$-defends g.

Our running example shows that, unfortunately, the basic extension no longer contains g_4 and g_5.

Example 4 (Travel, continued) Under the new definition, the first four realization trees g_1, g_2, g_3, g_4 with desire ja attack each other. There are four preferred extensions: $S_1 = \{g_1\}$, $S_2 = \{g_2\}$, $S_3 = \{g_3\}$, $S_4 = \{g_4, g_5\}$. The basic extension is $\emptyset$.

The reason that the basic extension no longer works as desired, is that multiple plans for the same desire attack each other. In such cases the basic extension does not contain any of these. This is a consequence of the fact that basic extensions are always unique. There are various ways to repair this. One approach is to define a notion of extension* which is constructive, like a basic extension, but which splits in case of multiple plans for the same desire. The following definition adds all arguments which are defended, and non-deterministically adds one argument that is only attacked by alternative plans for the same desire.

Definition 8 Let F be as in definition 6. Let $C(S) = \{g(d) \in G \backslash S \mid \forall g'(d') \in G$ if g' D-defend g then $d = d'\}$.

- S is a basic extension* iff it is a least fixpoint of the non-deterministic function
$$F'(S) = \begin{cases} F(S), & \text{if } C(S) = \emptyset, \\ F(S) \cup \{g\},\ g \in C(S) & \text{otherwise.} \end{cases}$$

There is always at least one basic extension*. In the running example, the basic extension* comes out as desired.

Example 5 (Travel, continued) The unique basic extension* is $\{g_4, g_5\}$.

It is disappointing to have to alter the argumentation framework after all, but a remaining advantage of our approach is that it becomes easier to combine planning with goal generation in the same framework. This is illustrated in the following section.

Goal Generation

Agent architectures often assume separate components for goal generation, goal selection and planning (Wooldridge 2002, p 76). In our terminology, goal generation produces potential goals on the basis of an agent's beliefs, desires (internal motivation), and possibly obligations (external motivation). Goal selection is the subsequent process of deciding which consistent subset of goals will be pursued. These selected goals become intentions. Planning is the process of finding a sequence of actions to achieve the intentions.

Thus we have to deal with two kinds of reasoning: forward reasoning from current beliefs to desirable states (deduction), and backward "means-ends" reasoning from desirable states to required actions (abduction). This combination is often problematic.

1. The goal generation and goal selection components partly depend on planning, because potential or selected goals are required to be feasible (Cohen & Levesque 1990; Rao & Georgeff 1998). A goal is feasible when some plan exists that is likely to achieve it. Rao and Georgeff call an agent that only generates feasible goals *strongly realistic*. It is waste of resources to consider infeasible goals. The feasibility restriction requires that planning can provide feedback to goal generation and selection.
2. The application of a priority order is difficult. The easiest solution is to define a local priority order over rules (Reiter 1980). However, single rules often have undesired consequences. Application of rules should therefore be compared by their outcomes, using a utility value for example. Other research on goal generation uses maximally consistent sets of rules, as the main interface between components (Broersen *et al.* 2001; 2002; Dastani & van der Torre 2002). This has some drawbacks, both practical and conceptual. The sets themselves become large, and their consistency becomes difficult to check and maintain.

To address these problems, we will again apply techniques from argumentation theory, combining goal generation with our version of Amgoud's framework to reason about conflicting desires and plans.

Extension with goal generation

Thus far, arguments are trees of realization for a single desire. In this section we generalize the notion of desire to desire rules, analogous the the rules used for plans and background knowledge, and we introduce the notion of a goal set: a set of mutually compatible desires. The definition of realization trees is adapted, and combines the derivation of a goal set with a plan to realize it.

Definition 9 extends Amgoud's notion of unconditional desires in definition 2 to a set of desire rules D, similar to the planning and background rule sets P and Σ. A desire rule $l_1 \wedge \ldots \wedge l_{n-1} \rightarrow l_n$ in D represents that l_n is desired in the context $l_1 \wedge \ldots \wedge l_{n-1}$, a planning rule represents that l_n is achieved if $l_1 \wedge \ldots \wedge l_{n-1}$ is achieved, and a background rule represents that l_n is true when $l_1 \wedge \ldots \wedge l_{n-1}$ is true. Since a decision variable needs no planning rules, we require that the head of a planning rule is not a decision variable.

Definition 9 Let A, N and L be as defined in definition 1. A *desire-plan description* is a tuple $\langle D, P, \Sigma \rangle$ with D, P and Σ sets of rules from L, such that the heads of rules in P are built from a variable in N.

Here are some examples of desire-plan descriptions. The following example is concerned with food and wine.

Example 6 (Dinner) Let $A = \{e, r, d, t, w\}$, $N = \emptyset$ represent a menu with e for entrecote, r for red wine, d for daurade, t for trout and w for white wine. Let dinner preferences be $D = \{e,\ t,\ e \rightarrow r,\ d \rightarrow w,\ t \rightarrow w\}$, $P = \emptyset$ and $\Sigma = \emptyset$. The agent desires red wine with meat, or white wine with fish.

The following example extends Amgoud's (2003) example 1.

Example 7 (Travel+) Let $A = \{$*ag, fr, hop, dr, sol, w, cp, gp*$\}$ and $N = \{$*wa, ja, hjor, dlc, fp, pa, t, vac*$\}$, with the interpretation:

wa	war in Africa	*ag*	go to the agency
ja	journey to Africa	*fr*	friend brings tickets
hjor	have job on return	*hop*	go to hospital
dlc	deadline close	*dr*	go to a doctor
fp	finish paper	*sol*	solicit for work
pa	paper is accepted	*w*	work
t	get the tickets	*cp*	call program chair
vac	be vaccinated	*gp*	write a good paper

Consider the following desire-plan description:

$D = \{\neg wa \rightarrow ja,\ ja \rightarrow hjor,\ dlc \rightarrow fp,\ fp \rightarrow pa\}$
$P = \{t \wedge vac \rightarrow ja,\ ag \rightarrow t,\ fr \rightarrow t,\ hop \rightarrow vac,$
$dr \rightarrow vac,\ sol \rightarrow hjor, w \rightarrow fp,\ cp \rightarrow pa,\ gp \rightarrow pa\}$
$\Sigma = \{\neg wa,\ dlc,\ w \rightarrow \neg ag,\ w \rightarrow \neg dr,\ w \rightarrow \neg hop\}$

In some contexts, the agent has a desire to journey, to have a job, to finish a paper and get this paper accepted. There are several ways to achieve these desires. For example, to get a paper accepted the agent can either write a good paper or call the program chair.

A goal set is a set of *related* desires, such that we build a plan for these related desires. But how can we formalize this notion of relatedness? One way is to incorporate a new notion in the desire-plan description, but that makes the system more complicated, and leaves a higher burden on the user of the system. We therefore use an implicit definition. A desire rule depends on another one, if the former can be applied after the second one, but the former cannot be applied when the latter has not been applied. The notion of application is defined by means of the intermediate notion of a goal tree, which is a tree that represents a derivation of goal sets from desire rules. A candidate goal tree contains all desire rules which can be applied, and goal trees are only those trees which contain related desires, i.e., all desire rules besides the desire rule in the root can only be applied due to the application of earlier desire rules.

Definition 10 Let $\langle D, P, \Sigma\rangle$ be a desire-plan description.

- A *conditional goal set* is a pair of sets of literals (B, H).
- A *candidate goal tree* for *candidate goal set* GS , written as $c(GS)$, is a finite linear tree consisting of conditional goal sets such that for each (B, H) we have a desire rule $l_1 \wedge \ldots \wedge l_n \rightarrow l \in D$ such that:
  (a) $B = \{l_1, \ldots, l_n\} \subseteq Cl(\Sigma, C)$, where C is the union of all literals occurring in the ancestors of (B, H).
  (b) if (B, H) is the root, then $H = \{l\}$, otherwise $H = \{l\} \cup H'$ when the unique parent of (B, H) is (B', H') for some B'.
- A *goal tree* for a goal set GS, written as $g(GS)$, iff it is a candidate goal tree for a candidate goal set and there is no set of candidate goal sets $\{GS_1, \ldots, GS_n\}$ with each $GS_i \neq GS$ and $GS = GS_1 \cup \ldots \cup GS_n$. A *maximal goal set* is a goal set which is maximal with respect to set inclusion.

The following proposition illustrates that constructing a goal tree is analogous to applying inference rules in classical logic

Proposition 2 (Goal set) We write $H(R)$ for the set of heads of rules in R. Given a desire-plan description $\langle D, P, \Sigma\rangle$, a finite set of literals GS is a *goal set* iff there exists a subset D' of D such that:

- $GS = Cl(D' \cup \Sigma) \cap H(D')$;
- $Cl(D' \cup \Sigma)$ is consistent, i.e., does not contain l and $\neg l$;
- There is no set of goal sets $\{GS_1, \ldots, GS_n\}$ with each $GS_i \neq GS$ and $GS = GS_1 \cup \ldots \cup GS_n$.

The following two examples illustrate that a goal set is a set of related desires.

Example 8 (Travel+, continued) The goal sets are $\{ja\}$, $\{ja, hjor\}$, $\{$*fp*$\}$ and $\{$*fp*$, pa\}$. The set of desires with $\{$*ja, hjor, fp, pa*$\}$ is not a goal set, because it can be split in $\{ja, hjor\}$ and $\{$*fp*$, pa\}$. The latter two are the maximal goal sets. Here, ja and $hjor$ are related, because the desire to have a job on return from travel (*hjor*) is conditional on making a journey to Africa (*ja*). Likewise, (*fp*) and (*pa*) are related, because the desire to have a paper accepted (*pa*) is conditional on finishing a paper before going to Central Africa (*fp*).

Example 9 (Dinner, continued) We have the following goal sets: $\{e\}$, $\{e, r\}$, $\{t\}$, $\{t, w\}$. The set of desires $\{e, r, t, w\}$ is not a goal set, because it can be split in $\{e, r\}$ and $\{t, w\}$. The sets $\{e, r\}$ and $\{t, w\}$ are the maximal goal sets. Here, for example e and r are related, because the desire for red wine (r) is conditional on having entrecote (e).

Now we define a goal realization tree as a combination of a goal tree with the realization trees of the goals. Because a goal tree is linear, there is only one leaf, which contains a goal set. A goal set represents a cluster of goals that belong together. The planning process produces realization trees for each of the literals in the goal set.

Definition 11 Given a desire-plan description $\langle D, P, \Sigma\rangle$, a *goal realization tree* is a goal tree $g(GS)$, which is connected for each literal $l \in GS$ to a realization tree $g(l)$.

Figure 2 shows two goal realization trees for example 7. The goal generation steps are visualized with downward arrows.

Just like realization trees, goal realization trees may conflict in various ways. To deal with such conflicts, we can use an argumentation framework which is basically the same as before, except that we have to adapt the exclusion of multiple plans for the same goal. The issue is that we may have goal sets which partially overlap, for example there may be three goal realization trees with goal sets $\{p, q\}$, $\{q, r\}$ and $\{r, p\}$. In that case we only want to include two of the three goal realization trees in an extension.

We solve this issue using two ideas. First we define an argument as a pair of a goal realization tree and a literal in the goal set of the goal realization tree. Second, we say that an argument attacks another argument when either there are complementary literals in the goal realization trees, or the goal literal of the latter occurs in the goal set of the former. We thus may have two goal realization trees in the same extension that contain some identical goal literals, but there always has to be at least one goal literal in each goal deliberation tree which does not occur in the goal sets of the other goal realization trees in the extension.

$$
\begin{tikzcd}[column sep=small]
 & & \langle\{\neg wa\},\{\mathit{ja}\}\rangle \arrow[d] & \\
 & & \langle\{\mathit{ja}\},\{\mathit{hjor},\mathit{ja}\}\rangle & \\
 & \langle \mathit{ja},\{t,\mathit{vac}\}\rangle \arrow[ur] & & \langle \mathit{hjor},\{\mathit{sol}\}\rangle \arrow[ul] \\
\langle t,\{\mathit{ag}\}\rangle \arrow[ur] & & \langle \mathit{vac},\{\mathit{hop}\}\rangle \arrow[ul] & \langle \mathit{sol},\{\}\rangle \arrow[u] \\
\langle \mathit{ag},\{\}\rangle \arrow[u] & & \langle \mathit{hop},\{\}\rangle \arrow[u] &
\end{tikzcd}
$$

$$
\begin{tikzcd}[column sep=small]
 & \langle\{\mathit{dlc}\},\{\mathit{fp}\}\rangle \arrow[d] & \\
 & \langle\{\mathit{fp}\},\{\mathit{pa},\mathit{fp}\}\rangle & \\
\langle \mathit{pa},\{\mathit{cp}\}\rangle \arrow[ur] & & \langle \mathit{fp},\{w\}\rangle \arrow[ul] \\
\langle \mathit{cp},\{\}\rangle \arrow[u] & & \langle w,\{\}\rangle \arrow[u]
\end{tikzcd}
$$

Figure 2: Two goal realization trees

Definition 12 Let $\langle G, Attack\rangle$ be an SHD for a desire-plan description $\langle D, P, \Sigma\rangle$, such that G contains all pairs $\langle g(GS), l\rangle$ such that $g(GS)$ is a goal realization tree and $l \in GS$. Let $S \subseteq G$ and $g, g_1, g_2 \in G$ be (sets of) such pairs.

- $\langle g_1(GS_1), l_1\rangle$ attacks $\langle g_2(GS_2), l_2\rangle$, iff either
  1. there exist actions a_1 and a_2 in the nodes of the goal realization trees of g_1 and g_2 respectively, such that a_1 and a_2 conflict, or
  2. $g_1 \neq g_2$ and $l_2 \in GS_1$: the literal of g_2 occurs in g_1's goal realization tree.

The other notions remain the same as in definition 7.

Further Research

We only considered two phase deliberation, consisting of goal generation, followed by planning. However, a plan can also trigger new desires and goals. For example, physical exercise may cause a desire to drink. Or a plan to visit Paris, may trigger a desire to read about Paris, even beyond the essential information for the trip. Such reading goals trigger further plans, to visit a library say, etc. Thus planning and goal generation may be intertwined. Our current research is concerned with formal ways to represent such intertwined goal realization trees. Our first results show that we we can use simple extensions of the notions developed in the previous sections. How to interpret such representations remains an open issue.

First we define an extended goal realization tree as a combination of several goal trees and trees of realization. We also have to relax the first condition in the definition of a (candidate) goal tree. We therefore introduce the notion of context in the definition of goal tree.

Definition 13 A *goal tree in context S* is defined analogous to a goal tree in definition 10, except that the first clause for candidate goal trees is replaced by the following one:

(a′) $B = \{l_1, \ldots, l_n\} \subseteq Cl(\Sigma, C)$, where C is the union of all literals occurring in the ancestors of (B, H) or in S.

We now define extended goal realization trees as sequences of goal tree - realization tree - goal tree - realization tree - etc. As before, we only want related desires to occur in the goal trees.

Definition 14 (Sketch) An *extended goal realization tree* is a set of goal trees in context and trees of realization, such that:

- the leaf of each goal tree $g(GS)$ is connected, for each literal $l \in GS$, to a realization tree $g(l)$.
- the root of each goal tree contains a set of leaves of realization trees (atomic actions) , and there is exactly one goal tree with an empty set;
- the context of each goal tree is the set of literals occurring in the realization trees (together with its ancestors) that its root is connected to.

Again the argumentation framework remains the same, with the minor adaptation that an argument is an extended goal tree together with a goal literal, such that the literal occurs in the goal set of any of the goal trees occurring in the extended goal realization tree.

Related Research

Despite the analogy between arguments and plans, we know of few other researchers apart from Amgoud, that have combined planning and argumentation theory.

There has been relevant research on the differences of deduction and abduction in knowledge representation and reasoning. Often deduction is associated with prediction, whereas abduction is associated with explanation (Shanahan 1989). We agree with this generalization, but note that it only refers to facts, which are beyond the control of an agent. For controllable variables or actions, goal-directed reasoning is the best explanation. Note furthermore that abduction is used in natural language understanding for what is called goal recognition: i.e., to infer the goal of the speaker from what was said.

A combination of deduction and abduction has been applied to agent-architectures before (Kowalski & Sadri 1999; Sadri & Toni 2000). In this architecture a logic program corresponds to beliefs, while integrity constraints can be interpreted as desires or reactive rules. A set of so called *abducibles* is used to expand a logical theory, with goals or plans, for example.

A combination of abduction and deduction can also be applied to agent interaction. Hindriks et al. (2000) use abduction to generate responses to queries, and Sadri et al. (2002) use it in the deliberation cycle for agents in a dialogue.

The importance of goal generation has been argued by Thomason (2000). Broersen et al. (2001; 2002) study an abstract architecture for goal generation, which distinguishes production rules for beliefs, obligations, intentions and desires. Different priorities among these sets of rules, correspond to different agent types, and produce different goal sets. Possibly, agent types can be extended to comprise different planning strategies as well, or even different strategies for combining goal generation and planning. This remains an interesting direction of further research

Concluding remarks

In this paper we consider the combination of planning and goal generation in an argumentation framework. Goals have plans to realize them, and are modeled just like claims that have arguments to support them.

With respect to the research questions of the introduction, we can now state the following.

1. Conflicts between default rules representing practical reasoning or desires, differ from conflicts between plans, because two plans for the same goal will conflict under the assumption that both should be executed. Such plans are competing for resources and deliberation effort.
2. We can represent conflicts between plans, by adapting the notion of attack to include competition between plans. By adapting the notion of attack, which is domain dependent anyhow, we can leave the rest of the argumentation framework intact.
3. The argumentation framework for conflicts between plans can be extended to deal with goal generation and conflicts between goals, by means of the introduction of goal trees and goal sets and by connecting them to realization trees.

Future research is concerned with a case study of a complex goal generation and planning problem. We hope to find practical guidelines for combining goal generation and planning, that can be captured by our theoretical framework. Moreover, we intend to extend the goal realization trees to sequences of goal trees and realization trees, such that plans can trigger new goals.

A first implementation of the definitions and examples has been provided in Prolog and is available online. For more information on the implementation, see `http://boid.info/boidarg/`.

Acknowledgements We thank Leila Amgoud for discussions on the topics raised while the authors visited the IRIT laboratory in March 2003 and September 2003. Thanks also to Mehdi Dastani for valuable input.

References

Amgoud, L., and Cayrol, C. to appear. On the use of an ATMS for handling conflicting desires. In *Proceedings of the Ninth International Conference on Principles of Knowledge Representation and Reasoning (KR'04)*. AAAI.

Amgoud, L. 2003. A formal framework for handling conflicting desires. In Nielsen, T. D., and Zhang, N. L., eds., *Proceedings of Seventh European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'03)*, LNCS 2711. Springer. 552–563.

Boella, G. 2002. Decision theoretic planning and the bounded rationality of BDI agents. In *In Proceedings of workshop on game theoretic and decision theoretic agents (GTDT'02)*, 1–10. AAAI Press.

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

Broersen, J.; Dastani, M.; Hulstijn, J.; and van der Torre, L. 2002. Goal generation in the BOID architecture. *Cognitive Science Quarterly* 2(3-4):431–450. special issue on *Desires, goals, intentions, and values: Computational architectures*.

Cohen, P., and Levesque, H. 1990. Intention is choice with commitment. *Artificial Intelligence* 42:213–261.

Dastani, M., and van der Torre, L. 2002. What is a normative goal? In Lindemann, G.; Moldt, D.; and Paolucci, M., eds., *Proceedings of International Workshop on Regulated Agent-Based Social Systems: Theories and Applications (RASTA'02)*, 210–227.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77:321–357.

Hindriks, K.; de Boer, F.; van der Hoek, W.; and Meyer, J.-J. 2000. Semantics of communicating agents based on deduction and abduction. In Dignum, F., and Greaves, M., eds., *Issues in Agent Communication*, LNAI 1916, 63–79. Springer Verlag.

Kowalski, R. A., and Sadri, F. 1999. From logic programming towards multi-agent systems. *Annals of Mathematics and Artificial Intelligence* 25(3-4):391–419.

Rao, A. S., and Georgeff, M. P. 1998. Decision procedures for BDI logics. *Journal of Logic and Computation* 8(3):293–343.

Reiter, R. 1980. A logic for default reasoning. *Artificial Intelligence* 13:81–132.

Sadri, F., and Toni, F. 2000. Abduction with negation as failure for active and reactive rules. In Lamma, E., and Mello, P., eds., *Advances in Artificial Intelligence: Proceedings of the 6th Congress of the Italian Association for Artificial Intelligence (AI*IA'99)*, volume LNCS 1792, 49–60. Springer Verlag.

Sadri, F.; Toni, F.; and Torroni, P. 2002. An abductive logic programming architecture for negotiating agents. In Flesca, S., et al., eds., *Proceedings of the European Conference on Logics in Artificial Intelligence (JELIA'02)*, LNAI 2424, 419 – 431. Springer Verlag.

Shanahan, M. 1989. Prediction is deduction but explanation is abduction. In Sridharan, N. S., ed., *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI'89)*, 1055–1060. Morgan Kaufmann.

Thomason, R. 2000. Desires and defaults: A framework for planning with inferred goals. In Cohn, A. G.; Giunchiglia, F.; and Selman, B., eds., *Proceedings of the 7th International Conference on the Principles of Knowledge Representation and Reasoning (KR'00)*, 702–713. Morgan Kaufmann, San Mateo, CA.

Wooldridge, M. 2002. *An Introduction To MultiAgent Systems*. John Wiley and Sons, Chchester.