Argumentation for Access Control

Guido Boella[1], Joris Hulstijn[2], and Leendert van der Torre[3]

[1] Universitá di Torino
[2] Vrije Universiteit, Amsterdam
[3] CWI Amsterdam and Delft University of Technology

Abstract. In this paper we are interested in argument based reasoning for access control, for example in the context of agents negotiating access to resources or web services in virtual organizations. We use a logical framework which contains agents with objectives concerning access to a resource or provision of a service, including security objectives. The access control mechanism is described by a set of policy rules, that specify that access to a resource or service requires a specific set of credentials. Our contribution is a formalization of the reasoning about access control using a planning theory formalized in Dung's abstract argumentation framework. We build on Amgoud's argumentation framework for plan arguments, which is based on an adaptation of Dung's notion of defence. Our formal argumentation framework allows arguments about the backward derivation of plans from objectives and policy rules (abduction), as well as arguments about the forward derivation of goals from general objectives. We show that reasoning about the feasibility of goals requires mixed goal-plan arguments, and we show how to formalize the plan arguments in Dung's framework without adapting the notion of defence.

1 Introduction

Traditionally, access control mechanisms are centered around the certification of identities, as for example in the X.509 protocol. When managing access control becomes more complex, and agents want to reason about access control, a *declarative approach* to the specification and verification of access control policies becomes necessary [3]. Declarative policy rules state which signed credentials are needed for actions. Moreover, with the advent of web services and virtual organizations, access control becomes not only based on the identity of a client, but on the credentials needed to access the service. Since the required credentials are not always known to the client, the process of accessing a resource becomes an interaction between client and server, discussing modalities until they reach an agreement. Thus, based on declarative policies, Koshutanski and Massacci [7, 8] developed *interactive access control.* A similar approach is advocated by Bonatti and Samarati [4].

In interactive access control approaches, a so called trust management system is introduced, which contains a logical reasoning engine to make it easier to reason about the details of an access request, and about the context in which access requests are evaluated. The application that makes the access request does not have to guess beforehand which credentials may be needed. Credentials are provided by the applicant and verified, using deduction. If the credentials are insufficient to satisfy the access control policy, abduction is used to work out which (minimal) set of credentials is missing. The server will then request the applicant to provide these missing credentials. This process is repeated until the access is granted or the applicant runs out of credentials. This explains the term interactive access control.

We are interested in the case in which parties negotiate to find an agreement about which policy to apply, because there may be more than one way to achieve a security objective. Parties use arguments to support their case until one wins the dispute. Moreover, during this negotiation process, further security objectives can arise. For example, once an agent has gained access to a resource, requests for another resource may have to be denied, to avoid potential conflicts of interest.

To model this situation we suggest the following conceptualization. The search for credentials can be modeled as a planning process, where presentation of the credentials corresponds to actions. A declarative access control policy corresponds to a skeleton plan. It consists of the combinations of credentials that are required to achieve some overall security objective. Thus different alternative combinations of credentials may exist to achieve the same objective. The set of objectives need not remain stable; as the environment changes, or as the underlying security preferences change, new objectives may be triggered. Thus we distinguish between two processes: *plan generation*, the derivation of combinations of credentials that will achieve those objectives, and *goal generation*, the derivation of security objectives on the basis of the state variables and basic security principles. This conceptualization is based on an analogy with non-classical planning, where goals are derived on the basis of more basic desires and beliefs about the current situation [15].

A formal framework for negotiation about access control needs (i) a formal representation of the content, in this case arguments about access control, and (ii) a formal representation of the possible moves (request, grant, challenge, defend, etc.) and a interaction protocol to specify what sequences of moves are well formed. In this paper we address only the former of these two issues: the the formalization of arguments about interactive and declarative access control situations. This breaks down into two sub-questions:

1. Which kind of arguments can be distinguished in interactive access control?
2. What is a logical framework for arguments about interactive access control?

We define arguments as trees that represent a line of reasoning. To distinguish the arguments in this theory, we distinguish goal and plan arguments for two relevant kinds of reasoning. Goal generation needs forward reasoning from the current state of affairs to preferable objectives (deduction). In case several sets of mutually compatible objectives are derived, so called options, a selection has to be made on the basis of some priority principle. Plan generation on the other hand, needs backward or "means-ends" reasoning from objectives to required and missing credentials (abduction). Moreover, we define mixed goal-plan arguments for the interaction between goal and plan generation. Goal generation and the subsequent process of selecting objectives partly depend on plan generation, because potential goals are required to be feasible. A goal or objective is called feasible when some policy exists that is likely to achieve it. In the context of agent planning Rao and Georgeff [11] call an agent that only generates feasible goals *strongly realistic*. It is waste of resources to consider infeasible goals. The feasibility restriction requires that plan generation can somehow constrain goal generation.

For the logical framework we apply techniques from formal argumentation theory, which allow us to cope with both the interaction and the planning aspects of interactive access control, and which also offers the advantages of non-monotonic reasoning. 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 [5] 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 [1, 2]. The central analogy is the following. An objective that has several possible plans, i.e. policies or combinations of credentials, can be modeled just like an argument which consists of a claim with the supporting argumentations. The attack relation defined over the set of arguments can serve as a criterium to deal with possible conflicts among policies, and to select a set of compatible policies that achieves a set of objectives. 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. This alteration is discussed in depth.

To illustrate the use of arguments for interactive access control, we present a running example throughout this paper.

The paper is structured as follows. In Section 2 we introduce the running example and explain formal arguments of policies and objectives. In Section 3 we show how to reason about these arguments in an argumentation framework. Related work and concluding remarks end the paper.

2 Arguments for policies and objectives

In interactive access control situations we consider the possibility that clients and servers have their own preferences and principles concerning which credential to disclose or require. Hence, parties may be said to argue about the outcome of an access request. We introduce a running example to illustrate this possibility. The following dialogue between a client A and a library clerk B could take place in a traditional library, but it is easy to imagine a similar exchange – formalized – at some automated online article repository. This example illustrates two aspects. First, different sets of credentials can be used to access a service. In our example, a pass, credit and a signed order are applied in turn. Secondly, during the interaction process new objectives can pop up. In our example the requirement to let successful applicants participate in a survey leads the librarian to request the applicant to fill in a questionnaire.

A: I would like to retrieve an article.
B: Yes, but you have to buy a subscription to the library
A: I am a University employee.
B: Please show me your pass.
A: $<$ showing pass $>$

B: You have to pay for the copy of the paper anyway.
A: Ok, I will use my credit.
B: Your credit has already depleted.
A: I will use a signed order from my boss, instead.
B: $<$ checking order $>$ Ok.
A: This is the article I would like to retrieve.
B: Here it is. Could you please fill out this questionnaire about the article?
B: Ok.

To formalize the example we need to introduce quite some notation. We need to represent credentials and facts about the world. More importantly, we need policy rules that indicate what credentials are needed to grant some access or service request. And finally, we need rules that indicate how additional objectives can be derived. In the logical language, we distinguish among credentials (C) and state variables (S).

Definition 1. Let C and S be two disjoint sets of credentials and state variables respectively. Let L be a propositional language built from $C \cup S$. A literal l is an element of C or S, 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. 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 $S^{i+1} = S^i \cup \{l \mid l_1 \wedge ... \wedge l_n \rightarrow l \in R,\ \{l_1 ... l_n\} \subseteq S^i\}$.

A so called *objective-policy description* consists of a set of conditional objectives (O), a set of policy rules (P), and a set of integrity constraints (K), representing general knowledge. A conditional objective $l_1 \wedge \ldots \wedge l_{n-1} \rightarrow l_n$ in O means that l_n is required in the context $l_1 \wedge \ldots \wedge l_{n-1}$; a policy rule means that l_n is achieved if $l_1 \wedge \ldots \wedge l_{n-1}$ is achieved, and an integrity constraint represents that l_n is true when $l_1 \wedge \ldots \wedge l_{n-1}$ are true. Since a credential needs no further policy rules, we require that the head of a policy rule is not a credential. In general, an objective is something which should be true but is not true yet, a policy rule explains how objectives can be achieved, and integrity constraints contain all other relevant rules. These rules behave as production rules. For explanatory reasons, we restrict the language here to conjunction and a syntactic form of negation. The general idea is to use these rules to generate extensions. An extension is a set of arguments or 'policies', depending on how you look at it, that are mutually compatible. Ideally, an extension will contain policies to fulfill all objectives, while no integrity constraints are violated. More elaborate logics exist in the literature on logic programming. Some of those are mentioned in section 4.

Definition 2. Let C, S and L be as defined in Definition 1. An *objective-policy description* is a tuple $\langle O, P, K \rangle$ with O, P and K sets of rules from L, such that the heads of rules in P are built from a variable in S.

We illustrate the three kinds of rules in an objective-policy description of our running example.

Example 1 (Access Control). Suppose you want to get an electronic copy of an article from the digital library. To do so you have to be authorized to use the library and you have to comply with the digital rights. Authorization is only given after paying a subscription to the library with your e-money, or after showing a university employee pass. To comply with the digital rights you have to pay with your e-money or present a signed order of your employer. Moreover, if you want to get an mp3 file, you also have to pay with e-money. Since you have a limited budget, using your e-money in this way prevents you from paying the rights for the paper. So these objectives conflict.

Granting access to an article is relevant only when there is a request by an applicant. Once the applicant gets the paper, the system wants to collect a survey, and to do so the applicant has to fill in and send a questionnaire. Moreover, if the system receives a request for an mp3 file, it will let the applicant access it if it complies with the policy. Once the applicant has access to the mp3 file, the system will want to improve its bandwidth. This objective is achieved when the applicant decreases the downloading speed or shares the file.

Let $C = \{es, sp, el, sr, em, ra, f, rm, ds, sm\}$ and $S = \{a, al, cr, m, cs, ib\}$, where

a	access article from digital library		
al	authorized to access library	*ra*	request article
es	subscribe to library with e-money	*cs*	collect survey
sp	send university employee pass	*f*	fill in questionnaire
cr	comply with digital rights	*rm*	request mp3 file
el	pay library with e-money	*ib*	improve bandwidth
sr	signed order by employer	*ds*	decrease download speed
m	access mp3 file	*sm*	share downloaded mp3 file
em	pay mp3 file with e-money		

Consider the following objective-policy description:

$O = \{ra \to a,\ a \to cs,\ rm \to m,\ m \to ib\}$

$P = \{al \wedge cr \to a,\ sp \to al,\ es \to al,\ sr \to cr,\ el \to cr,\ f \to cs, em \to m,\ sm \to ib,\ ds \to ib\}$

$K = \{ra,\ rm,\ em \to \neg es,\ em \to \neg el\}$

The system has the objective to grant access to an article or mp3 file, when requested ($ra \to a, rm \to m$). There are several ways to achieve both objectives, but if we add $em \to \neg sr$ as a rule to K, there is no way to achieve both.

The applicant requests an article or an mp3 file. When the system sends an article, it further requires to collect a survey, and when it sends an mp3 file it further requires to improve the bandwidth. There are several ways in which the applicant can comply with these objectives of the system. E.g., to participate in the survey, the applicant can fill in a questionnaire and send it to the system, and to improve bandwidth the applicant can reduce the downloading speed or share the file it just downloaded.

We now define our first kind of arguments about interactive access control, involving the derivation of security objectives on the basis of the state variables and basic security principles. We define arguments for a *goal set*. A goal set is an option that is selected to be enforced, and which is derived from a set of *related* objectives, such that we can find mutually compatible policies that can realize them. A goal argument is simply a sequence (linear tree) that represents a derivation of a goal set. Using goal arguments, we say that a conditional objective *depends* on another conditional objective, if the former can only be applied on the basis of the latter.

There is one technical complication which has to do with the minimality of arguments, sometimes these are called minimal arguments. In our argumentation theory where an argument involves the derivation of a set of formulas (goal set) from another set of formulas (objectives), we can consider a minimality restriction on either the goal set or on the variables that occur in the tree. We define our notion of minimality using the intermediate notion of a candidate goal tree for potential goal sets, which contains all conditional objectives which can be applied, based on application of earlier conditional objectives. Intuitively, a candidate goal argument is a goal argument, if its goal set cannot be split into a set of goal sets.

Definition 3. Let $\langle O, P, K\rangle$ be an objective-policy description.

- A *goal set* G is a set of literals.
- A *candidate goal argument* for *goal set* G, written $c(G)$, is a finite linear tree consisting of pairs of sets of literals with its unique leave (B, G) or any B, such that for each node (B, H) there exists a conditional objective $l_1 \wedge \ldots \wedge l_n \rightarrow l \in O$ such that:
  (a) $B = \{l_1, ..., l_n\} \subseteq Cl(K, U)$, where U 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 argument* for *goal set* G, written $g(G)$, is a candidate goal argument $c(G)$ such that there is no set of goal sets $\{G_1, ..., G_n\}$ with each $G_i \neq G$ and $G = G_1 \cup ... \cup G_n$. A *maximal goal set* is a goal set which has a goal argument and which is maximal with respect to set inclusion.
- We say that two goal arguments *conflict* if they contain nodes $\langle B_1, H_1\rangle$ and $\langle B_2, H_2\rangle$ such that $Cl(K \cup B1 \cup H_1 \cup B2 \cup H_2, \emptyset) \vdash \bot$, where $\bot$ stands for any contradiction.

The running example illustrates that a goal set is derived from a set of related objectives.

Example 2 (Continued). The goal sets are $\{a\}$, $\{a, cs\}$, $\{m\}$ and $\{m, ib\}$. The set $\{$*a, cs, m, ib*$\}$ is not a proper goal set, because it can be split in $\{a, cs\}$ and $\{m, ib\}$. The latter two are the maximal goal sets. Here, a and cs are related, because the objective to collect a survey (*cs*) is conditional on granting access (*a*). Likewise, (*m*) and (*ib*) are related, because the objective to improve bandwidth (*ib*) is conditional on access to an mp3 file (*m*).

The following proposition illustrates that constructing a goal argument is analogous to deductively applying inference rules in classical logic, together with the minimality constraint. Note that this is a recursive definition, but since the goal set decreases and its finiteness, the definition is well founded.

$$\begin{tikzcd}
 & \langle a, \{al, cr\}\rangle & & & \langle a, \{al, cr\}\rangle & & & \langle a, \{al, cr\}\rangle & & & \langle a, \{al, cr\}\rangle & & \\
\langle al, \{sp\}\rangle \arrow[ur] & & \langle cr, \{sr\}\rangle \arrow[ul] & \langle al, \{sp\}\rangle \arrow[ur] & & \langle cr, \{el\}\rangle \arrow[ul] & \langle al, \{es\}\rangle \arrow[ur] & & \langle cr, \{el\}\rangle \arrow[ul] & \langle al, \{es\}\rangle \arrow[ur] & & \langle cr, \{sr\}\rangle \arrow[ul] & \langle m, \{em\}\rangle \\
\langle sp, \emptyset\rangle \arrow[u] & & \langle sr, \emptyset\rangle \arrow[u] & \langle sp, \emptyset\rangle \arrow[u] & & \langle el, \emptyset\rangle \arrow[u] & \langle es, \emptyset\rangle \arrow[u] & & \langle el, \emptyset\rangle \arrow[u] & \langle es, \emptyset\rangle \arrow[u] & & \langle sr, \emptyset\rangle \arrow[u] & \langle em, \emptyset\rangle \arrow[u] \\
 & t_1 & & & t_2 & & & t_3 & & & t_4 & & t_5
\end{tikzcd}$$

Fig. 1. The plan arguments for Example 1

Proposition 1 (Goal Set with goal argument). We write $H(R)$ for the set of heads of rules in R. Given an objective-policy description $\langle O, P, K\rangle$, a finite set of literals G is a *goal set* with a goal argument iff there exists a subset O' of O such that:

- $G = Cl(O' \cup K, \emptyset) \cap H(O')$;
- $Cl(O' \cup K, \emptyset)$ is consistent, i.e., does not contain l and $\neg l$;
- There is no set of goal sets with goal arguments $\{G_1, ..., G_n\}$ with each $G_i \neq G$ and $G = G_1 \cup ... \cup G_n$.

The second type of argument we consider are plan arguments constructed from policies, which serve as a way to represent and reason with policies and their motivating objectives. Each node of the tree represents a pair of a literals and a set of literals. Note that this is distinct rom the nodes in a goal arguments, where nodes are pairs of sets of literals. These nodes are interpreted as atomic plans to be executed for having access, are defined as tuples $\langle h, H\rangle$ analogous to a claim for h with support H [1,5]. The leaves of the tree are credentials.

Definition 4. A *plan argument* for $\langle O, P, K\rangle$ for a goal in a goal set $g \in G$, written $t(g)$, is a finite tree whose nodes are pairs of a literal and a set of literals, either $\langle h, \emptyset\rangle$ for any $h \in C$, called a credential; or $\langle h, \{l_1, ..., l_n\}\rangle$ for any rule $l_1 \wedge ... \wedge l_n \rightarrow h \in P \cup K$, such that

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

We say that two plan arguments conflict if they contain nodes $\langle h_1, H_1\rangle$ and $\langle h_2, H_2\rangle$ such that $Cl(K \cup H_1 \cup H_2, \{h_1, h_2\}) \vdash \bot$.

The plan arguments generated by Example 1 are shown in Figure 1. We visualize plan arguments with arrows directed from the leaves to the root.

Example 3 (Continued). There are five plan arguments, four for the objective *a*, and one for the objective *m*. We call the plan arguments for *a* t_1, t_2, t_3 and t_4, based on respectively $\{sp, sr\}$, $\{sp, el\}$, $\{es, el\}$ and $\{es, sr\}$. The plan argument for *m* is called t_5, and is based on $\{em\}$.

$$\begin{tikzcd}
 & & \langle\{ra\},\{a\}\rangle \arrow[d] & & & \langle\{rm\},\{m\}\rangle \arrow[d] & \\
 & & \langle\{a\},\{cs, a\}\rangle & & & \langle\{m\},\{ib,m\}\rangle & \\
 & \langle a,\{al,cr\}\rangle \arrow[ur] & & \langle cs,\{f\}\rangle \arrow[ul] & \langle ib,\{sm\}\rangle \arrow[ur] & & \langle m,\{em\}\rangle \arrow[ul] \\
\langle al,\{sp\}\rangle \arrow[ur] & & \langle cr,\{sr\}\rangle \arrow[ul] & \langle f,\{\}\rangle \arrow[u] & \langle sm,\{\}\rangle \arrow[u] & & \langle em,\{\}\rangle \arrow[u] \\
\langle sp,\{\}\rangle \arrow[u] & & \langle sr,\{\}\rangle \arrow[u] & & & &
\end{tikzcd}$$

Fig. 2. Two goal-plan arguments

The third type of argument we consider is called *goal-plan argument* and combines a goal argument with plan arguments for each of the goals in the goal set. Because a goal argument is a linear tree, there is only one leaf, which contains the goal set. A goal set represents a cluster of objectives that belong together. The policy generation process produces plan arguments, i.e. combinations of policies, for each of the objectives in the goal set.

Definition 5. Given an objective-policy description $\langle O, P, K\rangle$, a *goal-plan argument* is a goal argument $g(G)$, which is linked to a plan argument $t(l)$ for each literal $l \in G$.

Figure 2 shows two goal-plan arguments for Example 1. The goal generation steps are visualized with downward arrows.

In principle more complex arguments can be constructed, in which plan arguments are connected again to goal arguments. Such complex arguments are more difficult to handle, and therefore left for further research. We now turn to the argumentation theory to deal with these kinds of arguments.

3 Argumentation theory for interactive access control

In this section we develop a formal argumentation theory for our arguments for interactive access control. We use Dung's abstract framework, in which an argumentation framework is defined as a set of arguments with a binary relation that represents which arguments attack which other arguments [5]. In general, the attack relation is the parameter that covers domain dependent aspects of argumentation. In our case, the arguments are goal-plan arguments, and the attack relation has to be derived from conflicts between arguments [1].

Definition 6. An argumentation framework is a tuple $\langle T, Attack\rangle$ such that T is a set of arguments and $Attack$ is a binary relation over T.

At this point, it may seem that we can simply define the attack relation among goal-plan arguments by saying that argument 1 attacks argument 2 when argument 1 and 2 are conflicting. However, Amgoud [1] shows that such a symmetric attack relation leads to counterintuitive results. In her theory of plan arguments the semantics of a plan argument is a complete plan to achieve an objective. The aim is to achieve a maximum of objectives. If a given objective o_1 can be achieved with a plan p_1 then if another plan p_2 for the same objective attacks a plan p_3 of another objective o_2, we will accept p_3 to enable the agent to achieve its two objectives. In the running example in Figure 1, read $o_1 = a$, $p_1 = t_4$, $p_2 = t_1$, $o_2 = m$, and $p_3 = t_5$. Therefore, Amgoud proposes to alter the definition of defend since there is a difference between conflicts among the usual kinds of arguments, i.e., defaults or beliefs, and conflicts among plans.

We consider a different notion of attack. The basic idea is as follows. We want to achieve a maximal number of objectives, and consequently an extension contains a maximal number of goal-plan arguments. However, considering different policies for the same objective may burden the client with requests for unnecessary credentials, which it could prefer not to disclose for privacy reasons. We therefore exclude the possibility that an extension contains two goal-plan arguments for the same objective. In a sense, this is the opposite of Amgoud's approach. Under her definition a basic extension, for example, may contain several plan arguments for the same objective.

There is one complication due to our introduction of goal arguments. With only plan arguments, each argument is for a single goal literal. However, with goal arguments, arguments can be for a *goal set*, and such sets may overlap. For example, consider three goal-plan arguments for goal sets $\{p, q\}$, $\{q, r\}$ and $\{r, p\}$. In that case we only want to include two of the three goal-plan arguments in an extension. We deal with this issue by two ideas. First we define an argument as a pair of a goal-plan argument and a literal in the goal set of the goal-plan argument. Secondly, we say that an argument attacks another argument when either there are complementary literals in the goal-plan trees, or the objective literal of the latter occurs in the goal set of the former.

Our approach can be motivated as follows. We model the deliberation process of an access control manager. During goal generation, alternative policies for an objective are considered, to test the feasibility of the objective. Objectives 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 objectives, along with their policies. Thus an extension models a potential set of related objectives: an option. Ultimately, an access control manager must select one option to be enforced. Therefore it does not make sense to also consider alternative options within an extension. Keeping alternative options open requires additional deliberation effort. Moreover, policies may compete for resources and thus be incompatible.

The definitions of attack free, defend, preferred extension and basic extension are taken from Dung's framework.

Definition 7. An argumentation framework $\langle T, Attack \rangle$ for a objective-policy description $\langle O, P, K \rangle$ is an argumentation framework in which T contains all pairs $\langle t(G), l \rangle$ such that $t(G)$ is a goal-plan argument and $l \in G$. Let $S \subseteq T$ and $t, t_1, t_2 \in T$ be (sets of) such pairs.

- $\langle t_1(G_1), l_1 \rangle$ attacks $\langle t_2(G_2), l_2 \rangle$, iff either
  1. there exist two nodes p_1 and p_2 in the goal-plan arguments of t_1 and t_2 respectively, such that p_1 and p_2 conflict, or
  2. $t_1 \neq t_2$ and $l_2 \in G_1$: the literal of t_2 occurs in t_1's goal-plan tree.
- S is attack free iff there are no $t_1, t_2 \in S$ such that t_1 attacks t_2.

- S defends t iff for all $t_1 \in G$ such that t_1 attacks t, there is an alternative $t_2 \in S$ such that t_2 attacks t_1.
- 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 defend all their elements.
- S is a basic extension iff it is a least fixpoint of function $F(S) = \{t | t \text{ defended by } S\}$

Our running example shows that, unfortunately, the basic extension no longer contains t_1 and t_5.

Example 4 (Continued). Under the new definition, the plan arguments t_1, t_2, t_3, t_4 for objective a attack each other. There are four preferred extensions: $S_1 = \{t_2\}$, $S_2 = \{t_3\}$, $S_3 = \{t_4\}$, $S_4 = \{t_1, t_5\}$. The basic extension is $\emptyset$.

The reason that the basic extension no longer works as desired, is that multiple policies for the same objective 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 policies for the same objective. 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 objective.

Definition 8. Let $C(S) = \{t(o) \in T \backslash S \mid \forall t'(o') \in T \text{ if } t' \text{ defend } t \text{ then } o = o'\}$ and let F be as in Definition 7.

- S is a basic extension* iff it is a least fixpoint of the non-deterministic function
$$F'(S) = \begin{cases} F(S), & \text{if } C(S) = \emptyset, \\ F(S) \cup \{t\},\ t \in C(S) & \text{otherwise.} \end{cases}$$

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

Example 5 (Continued). The unique basic extension* is $\{t_1, t_5\}$.

4 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 [14]. A combination of deduction and abduction has been applied to agent-architectures before [9, 12]. In that case, deduction is used for reasoning with integrity constraints; abduction finds those actions or subgoals, that are required to achieve some goal.

A combination of abduction and deduction can also be applied to agent interaction. Hindriks et al. [6] use abduction to generate responses to queries, and Sadri et al. [13] use it in the deliberation cycle for agents in a dialogue.

Our application of goal generation and planning to interactive access control owes much to Koshutanski and Massacci [7, 8]. They too apply both deduction and abduction. Deduction is used to verify that a set of credentials would satisfy a request for access control, given a set of policy rules. Similar to planning, abduction is used to find the missing credentials to satisfy some request r. Suppose C_P is the set of current credentials, expressed as literals, and P_A is the set of policy rules, expressed as a logic program. Now use abduction to find a minimal set of missing credentials C_M, such that both $P_A \cup C_P \cup C_M \models r$ and $P_A \cup C_P \cup C_M \not\models \bot$. If no such set exists, access is denied. Otherwise the set C_M is sent as a response to the client, after which the process is repeated. Our contribution to this line of work, apart from the application of argumentation theory, is that we allow derivation of additional objectives on the basis of an earlier set of objectives. As the access control process becomes interactive and even argumentative, handling such additional requirements becomes necessary.

5 Concluding remarks

In this paper we address the formalization of interaction and argumentation about access control and service provision. Interactive argumentation about access control requires on the one hand a representation of the moves and the interaction protocol, and on the other hand a representation of the content: a logical framework to express argumentations about access control. We develop a logical framework for access control in which policies are described by sets of credentials, and two kinds of rules: conditional objectives that tell us when new objectives are created, and policy rules that tell us which credentials are needed to achieve these objectives.

In this logical framework we define arguments for access control as complex trees that relate credentials to objectives. Putting these arguments in Dung's abstract argumentation framework, we can use the attack relation derived from conflicts between basic policies and credentials, to derive extensions. The traditional ways of comparing extensions, e.g. preferred extensions, can now be reapplied. An extension corresponds to a set of related objectives along with the mutually compatible policies that realize them. In this way, a trust management system that must choose between different possibly conflicting security objectives, can now do so using the standard techniques of argumentation theory.

A subject of further research is a more detailed conflict resolution for policy conflicts. In selection of objectives, the application of a priority order is difficult. The easiest solution is to define a local priority order over rules. However, single rules often have undesired consequences. Rules should therefore be compared by their outcomes, using a utility value for example. Other research makes use of maximally consistent sets of rules to represent a set of related objectives. This has some drawbacks, both practical and conceptual. The sets themselves become large, and their consistency becomes difficult to check and maintain.

References

1. Amgoud, L. A formal framework for handling conflicting desires. In *Procs. of EC-SQARU'03*, LNCS 2711, pages 552–563. Springer, 2003.
2. Amgoud, L. and Cayrol, C. On the use of an ATMS for handling conflicting desires. In *Procs of KR'04*. AAAI, 2004.
3. Blaze, M., Feigenbaum, J., and Lacy, J. Decentralized trust management. In *IEEE Symposium on Security and Privacy*, pages 164–173. IEEE, 1996.
4. Bonatti, P. and Samarati, P. A uniform framework for regulating service access and information release on the web. *Journal of Computer Security*, 10(3):242–272, 2002.
5. Dung, P. M. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.
6. Hindriks, K., de Boer, F., van der Hoek, W., and Meyer, J.-J. Semantics of communicating agents based on deduction and abduction. In *Issues in Agent Communication*, LNAI 1916, pages 63–79. Springer Verlag, 2000.
7. Koshutanski, H. and Massacci, F. Interactive trust management and negotiation scheme. In *Procs.of FAST'04 Workshop*, pages 139–152. Kluwer, 2004.
8. Koshutanski, H. and Massacci, F. A system for interactive authorization for business processes for web services. In *Procs. of ICWE04*, LNCS 3140, pages 521–525. Springer Verlag, 2004.
9. Kowalski, R. A. and Sadri, F. From logic programming towards multi-agent systems. *Annals of Mathematics and Artificial Intelligence*, 25(3-4):391–419, 1999.
10. Kraus, S., Sycara, K., and Evenchik, A. Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence*, 104(1-2):1–69, 1998.
11. Rao, A. S. and Georgeff, M. P. Decision procedures for BDI logics. *Journal of Logic and Computation*, 8(3):293–343, 1998.
12. Sadri, F. and Toni, F. Abduction with negation as failure for active and reactive rules. In *Procs. of AI*IA'99*, volume LNCS 1792, pages 49–60. Springer Verlag, 2000.
13. Sadri, F., Toni, F., and Torroni, P. An abductive logic programming architecture for negotiating agents. In *Procs. of JELIA'02*, LNAI 2424, pages 419 – 431. Springer Verlag, 2002.
14. Shanahan, M. Prediction is deduction but explanation is abduction. In *Procs. IJCAI'89*, pages 1055–1060. Morgan Kaufmann, 1989.
15. Thomason, R. Desires and defaults: A framework for planning with inferred goals. In *Procs. of KR'00*, pages 702–713, 2000.