

# Argumentation Theoretic Foundations for Abstract Dependence Networks

Patrice Caire<sup>1</sup>, Leendert van der Torre<sup>1</sup>, and Serena Villata<sup>2</sup>

<sup>1</sup> SnT and CSC, University of Luxembourg, Luxembourg  
patrice.caire@uni.lu, leon.vandertorre@uni.lu

<sup>2</sup> INRIA Sophia Antipolis, France  
serena.villata@inria.fr

**Abstract.** In this paper we show how to argue about agreements based on dependence. First, we introduce a formal theory of arguing about agreements by instantiating Dung’s abstract theory of argumentation with proposals for agreements represented as dependence networks. Second, we show that acceptable agreements are exchange based—satisfying the so-called do-ut-des principle—and not redundant. Third, to further decrease the number of proposals, we define a notion of minimal proposals. Roughly, all proposals can be split into a number of minimal sub-proposals such that if the proposal is acceptable, then its minimal sub-proposals are acceptable too. We show that minimal proposals satisfy the indecomposable do-ut-des property, i.e., they cannot be split into two nonempty sub-proposals with at most one shared agent.

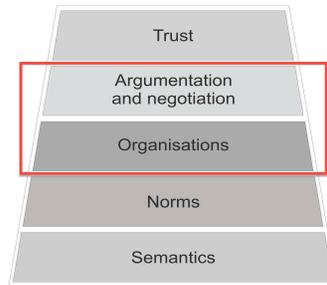
## 1 Introduction

Work in argumentation considers the problem of arguing about the formation of coalitions of agents [1, 11, 7, 8, 15], and reaching agreements [20, 3]. To the best of our knowledge there is no argumentation theory based on the theory of dependence, as developed for example by Castelfranchi and colleagues [14, 33, 31]. However, such theories are widely used to reach agreements among agents, for example in agreement technologies [4].

Billhardt *et al.* [4] envision that methods and mechanisms from the fields of semantic alignment, norms, organization, argumentation and negotiation, as well as trust and reputation are part of a “sandbox” to build software systems based on a technology of agreement. Starting with a well known definition of coordination as management of dependencies between organizational activities [21], they distinguish the detection of dependencies from taking a decision on which coordination action to apply. Their call-by-agreement interaction method first establishes an agreement for action, and the actual enactment of the action is requested thereafter. The normative context determines rules of the game, i.e. interaction patterns and additional restrictions. The so-called Agreement Technologies (AT) tower<sup>1</sup> of semantic alignment, norms, organization, argumentation, negotiation, trust and reputation is visualized in Figure 1.

---

<sup>1</sup> <http://bit.ly/HandbookAT>



**Fig. 1.** The AT tower [4] highlighting the layers we deal with.

Semantic technologies form the basis to deal with semantic mismatches and alignment of ontologies to give a common understanding of norms or agreements, defining the set of possible agreements. Norms and organizations determine constraints that the agreements, and the processes to reach them, have to satisfy. Organizational structures define the capabilities of the roles and the power and authority relationships among them. Argumentation and negotiation methods are used to make agents reach agreements. The agents use trust mechanisms that summarize the history of agreements and subsequent agreement executions in order to build long-term relationships between the agents. Billhardt *et al.* emphasize that these methods should not be seen in isolation, as they may well benefit from each other.

We are interested in the challenge of *combining* existing reasoning methods for semantic alignment, norms, organizations, argumentation, and trust. This is where abstract theories are promising. Dung [16] introduces abstraction in argumentation, which is a very useful concept to combine reasoning using instantiation. For example, it has been suggested that dependence networks can be seen as abstractions from BDI models [5, 6]. We foresee that the logical framework for agreement technologies will lead to a general framework of abstraction, refinement and instantiation of reasoning methods.

**Research question:** How to argue about dependence based agreements?

For argumentation we use the abstract theory of Dung [16], because it is a *de facto* standard and its abstract nature makes it well suited to be combined with a theory of agreements. The research question breaks down into the following sub-questions.

1. How to define the semantics of dependence networks as proposals for agreements?
2. How to discard proposals that will never be accepted?
3. How to restrict argumentation to minimal proposals without losing possible agreements?

For proposals we use a theory of dependence [14] representing power relations, where the power of agent  $p$  over agent  $d$  is represented by the dependence of agent  $d$  on agent  $p$ . Sauro [28] defines the *do-ut-des* property, based on a balance between the advantages and the burdens of the agents involved in a coalition, and the *indecomposable*

do-ut-des (*i-dud*) property taking into account also the costs and the risks deriving from the coalition formation process. The modeling framework  $i^*$  [33] and the software development methodology Tropos [10] introduce three kinds of dependencies, for goals, tasks and resources. We call them *reasons*.

A motivation for our work is the development of a theory of agreement technologies, where agreement-based coordination [4] is based on the management of dependence [22]. The agreement technologies tower (*Semantics, Norms, Organizations, Argumentation and Negotiation, Trust*) clearly distinguishes between making proposals for agreements and the *arguing* about these proposals to actually form agreements. The layers of the stack are often addressed as independent issues. In this paper, we present a first step towards the “unification” of the *Argumentation* and the *Organization* layers, where dependence networks are used to represent proposals for agreements for agent coalitions, fundamental to define a framework unifying all layers.

The aim of the paper is not argumentation-based negotiation [25], but it is to bridge two layers of AT stack, and make advancements in dependence networks research. More precisely, we start from dependence networks, which is a widely applied framework but either not very formal, and we bring new insights from the theory (*do-ut-des, i-dud*)<sup>2</sup> to a level where it is easier to catch. Moreover, we do not just define these concepts, but we provide argumentation theoretic foundations for them.

The assumptions we make in this paper are that the reasons are specific to the depender agent, that we do not represent so-called AND-dependence, and that payoffs of agents are sums of costs and benefits. These assumptions facilitate the presentation in this paper, and the proofs of its results.

The layout of the paper is as follows. We first introduce the theory of abstract dependence networks (Section 2), and then we show how to reason over acceptable and minimal proposals for agreements (Section 3) by combining the theory of abstract dependence networks with Dung’s theory of abstract argumentation. In Section 4 we define preferred, complete and grounded proposals.

## 2 Dependence theory

A dependence network is a social network, in which the relation among agents represents dependence.

### 2.1 Abstract dependence networks

Castelfranchi and colleagues [14, 31, 10] develop a theory of dependence networks, applicable to most social interactions. The abstract dependence networks we use in this paper are directed graphs labeled with three kinds of reasons [33]: the fulfillments of *goals*, the execution of *tasks*, and the production of *resources*. We follow Sauro [28] in that every reason can occur only in the dependence relations of a single depender. In contrast to Sauro, we represent only OR-dependence, no AND-dependence, such that

---

<sup>2</sup> None of existing works on argumentation-based negotiation studied concepts like *do-ut-des* or *i-dud*.

we do not consider jointly performed tasks, agents together fulfilling a goal, or together providing a resource. In other words, we assume that tasks, goals and resources can be performed, fulfilled or provided by a single agent.

**Definition 1 (Abstract dependence network).** An abstract dependence network  $N$  is a tuple  $\langle S, R, D \rangle$  where  $S$  and  $R$  are two disjoint sets (of agents and reasons, respectively), and  $D \subseteq S \times S \times R$  is a binary relation over agents for each reason such that  $(d_1, p_1, r)$  and  $(d_2, p_2, r)$  implies  $d_1 = d_2$ . If  $(d, p, r) \in D$ , then we say that agent  $d$  depends on agent  $p$  for reason  $r$ , or that agent  $p$  has power over agent  $d$  due to reason  $r$ . We call agent  $d$  the *depender*, and agent  $p$  the *performer of the dependence*.

Example 1 illustrates that an agent may depend on several other agents for the same reason, known as OR-dependence, meaning that she can choose *one* of these agents to make an agreement.

*Example 1.* Consider the abstract dependence network visualized in Figure 2.a. As common in this literature, a dependence of  $d$  on  $p$  for  $r$  is visualized by an arrow from  $d$  to  $p$  labeled with  $r$ . For example, if the reasons represent resources, then the flow of resources is inverse to the direction of the arrows. Dependence  $(b, a, r_1)$  may be read as “agent  $b$  depends on the Amazon service  $a$  to receive a book,”  $(b, f, r_1)$  as “agent  $b$  depends on the FNAC service  $f$  to receive a book,” and  $(a, b, r_2)$  as “the Amazon service  $a$  depends on agent  $b$  to receive a payment in exchange of the book.” The reason  $r_1$  can be a goal of the agent  $b$  to receive the book, or a task of Amazon or FNAC to send the book, or the book can be a resource. Summarizing, the agent can receive the book either from Amazon or from FNAC, but it is not able to pay FNAC for its services, only Amazon.

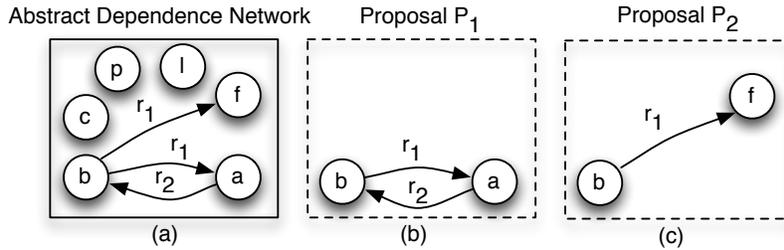


Fig. 2. A proposal is a dependence relation.

## 2.2 Proposals

We associate sets of proposals for agreements with dependence networks. A *proposal* for an agreement is a dependence relation, and a *proposal function* is a function from abstract dependence networks to sets of proposals. Dependence  $(d, p, r)$  proposes a commitment of the performer  $p$  to act in the benefit of the depender  $d$  to fulfill a goal, perform a task, or provide a resource.

**Definition 2 (Proposal function).** Let  $N = \langle S, R, D \rangle$  be an abstract dependence network. A proposal  $P \subseteq D$  for  $N$  is a dependence relation. A proposal function  $p$  is a function from abstract dependence networks to sets of proposals  $p(\langle S, R, D \rangle) \subseteq 2^D$ .

Note the similarity between our proposal function and Dung’s argumentation semantics [16], which we explain in Section 3.2. Argumentation semantics associate sets of nodes with a graph, we associate sets of edges with a (labeled) graph. In this sense, we may say that the proposal function gives *semantics* to the dependence networks.

Example 2 illustrates the difference between a dependence as part of a network, and a dependence as part of a proposal. Being part of a network does not imply a commitment of the agents to act according to the dependence.

*Example 2 (Continued).* Figures 2.b and 2.c visualize proposals  $P_1$  and  $P_2$ , respectively. Note that the dependence network is visualized as a straight box, and proposals with a dashed box. Figure 2.b visualizes a proposal where Amazon commits to send the book and agent  $b$  commits to pay for it, and Figure 2.c visualises a proposal where FNAC commits to send the book, but agent  $b$  does not commit to pay for it. A proposal function  $p$  may return multiple proposals, such as  $p(N) = \{P_1, P_2\}$ , on which the agents can argue.

A sub-proposal is a subset of a dependence relation.

**Definition 3 (Sub-proposal).** A proposal  $P_1$  is a sub-proposal of proposal  $P_2$  if and only if  $P_1 \subseteq P_2$ .

Example 3 illustrates a proposal with two sub-proposals.

*Example 3.* Consider the proposals visualized in Figure 3. Proposal  $P_1$  has two sub-proposals  $P_2$  and  $P_3$ . The agent  $b$  depends on Amazon to receive the book, but the payment is either via PayPal or via a credit card company, or both. Dependence  $(p, b, r_3)$  may be read as “PayPal  $p$  depends on agent  $b$  to confirm purchase of the book,” and  $(c, b, r_6)$  as “credit card company  $c$  depends on agent  $b$  to enter billing and shipping information.” The agents may argue which of  $P_1$ ,  $P_2$  or  $P_3$  is a better proposal.

### 3 Arguing about proposals

The first challenge is how to define the attack relation over proposals. It encodes dominance or preference over proposals, and from arguing about preferences by Visser *et al.* [32] it is known that such an attack relation may be acyclic and transitive, questioning the use of argumentation theory as a mechanism to decide conflicts among proposals.

#### 3.1 Agent behaviour

For the individual agents, we use a standard cost-benefit analysis. Each dependence comes with a cost for the performer, and a benefit for the depender, which are positive real numbers. The payoff for an agent is the sum of the benefits of each reason where he depends, minus the sum of the costs of the reasons where he performs. An investigation into uncertainty, utility, and the dependence among reasons, are left for future research.

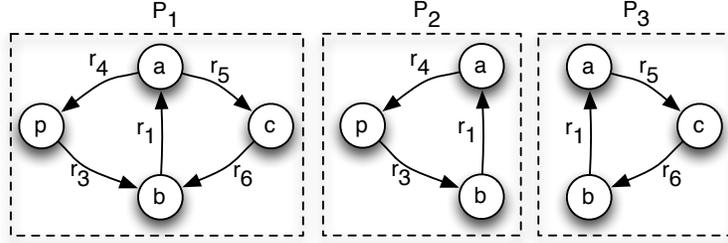


Fig. 3. Two sub-proposals  $P_2$  and  $P_3$  of  $P_1$ .

**Definition 4 (Cost-benefit).** Cost and benefit functions  $c, b : A \times R \rightarrow \mathbb{R}^+$  are functions from agents and reasons to positive real numbers. The reasons where agent  $a$  has a benefit, written as  $\text{benefits}(a, P)$ , are  $\{r \mid p \in A, r \in R, (a, p, r) \in P\}$  and the reasons where he has a cost, written as  $\text{costs}(a, P)$ , are  $\{r \mid d \in A, r \in R, (d, a, r) \in P\}$ . The payoff of a proposal  $P$  for agent  $a$ , written as  $\text{pay-off}(a, P)$ , is

$$\sum_{r \in \text{benefits}(a, P)} b(a, r) - \sum_{r \in \text{costs}(a, P)} c(a, r)$$

For modeling the interaction among the agents, we use a standard definition from game theory [23]. A proposal  $P_1$  dominates a proposal  $P_2$  if and only if for all agents involved in  $P_1$ , the pay-off in  $P_1$  is at least as good as in  $P_2$ , and for at least one of them it is strictly better. The restriction to agents of  $P_1$  is crucial: there may be agents who are worse off in  $P_2$ , but since these agents do not play a role in  $P_1$ , they cannot argue against it. Furthermore, there is no double counting for one and the same reason.

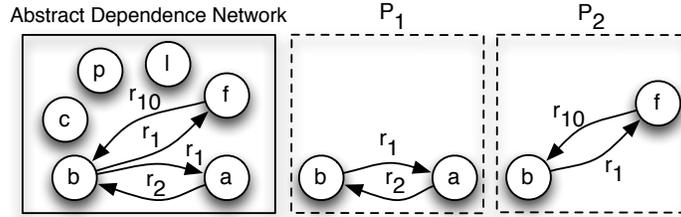
**Definition 5 (Dominance).** Let the agents of a proposal  $P$  be  $A(P) = \{a \in A : \exists(d, a, r) \in P \text{ or } \exists(a, p, r) \in P\}$ . A proposal  $P_1$  dominates proposal  $P_2$  if and only if we have  $\forall a \in A(P) : \text{payoff}(a, P_1) \geq \text{payoff}(a, P_2)$  and in addition  $\exists a \in A(P) : \text{payoff}(a, P_1) > \text{payoff}(a, P_2)$ .

Dominance is not strong enough to reject a proposal, as several proposals can be accepted at the same time. We introduce now an *attack* relation among proposals such that a proposal attacks another proposal when accepting the former implies that the latter is not acceptable. Moreover, we say that a proposal attacks itself if the payoff for at least one of the agents is negative.

**Definition 6 (Attack).** A proposal  $P_1$  attacks another proposal  $P_2$ , written as  $P_1 \prec P_2$ , if and only if  $P_1$  dominates  $P_1 \cup P_2$ , or  $P_1 = P_2$  and the payoff of at least one of the agents is negative.

Example 4 illustrates that the attack relation is not acyclic, and not necessarily transitive.

*Example 4 (Continued from Example 2).* Consider the abstract dependence network visualized in Figure 4, where agent  $b$  is able to pay for the FNAC service as well as for Amazon. This leads to two proposals  $P_1$  and  $P_2$  attacking each other.



**Fig. 4.** Attack cycle:  $P_1$  attacks  $P_2$  and vice versa.

### 3.2 Acceptable proposals

To define acceptable proposals, we introduce the basic concepts of Dung's abstract argumentation [16].

**Definition 7 (Argumentation semantics).** A proposal argumentation framework (AF) is a pair  $\langle \mathcal{P}, \hookrightarrow \rangle$  where  $\mathcal{P}$  is a set of proposals called arguments and  $\hookrightarrow \subseteq \mathcal{P} \times \mathcal{P}$  is the binary attack relation over proposals. Let  $C \subseteq \mathcal{P}$ . A set  $C$  is conflict-free if and only if there exist no  $P_i, P_j \in C$  such that  $P_i \hookrightarrow P_j$ . A set  $C$  defends an argument  $P_i$  if and only if for each argument  $P_j \in \mathcal{P}$  if  $P_j$  attacks  $P_i$  then there exists  $P_k \in C$  such that  $P_k$  attacks  $P_j$ . Let  $C$  be a conflict-free set of arguments, and let  $\mathcal{D} : 2^{\mathcal{P}} \mapsto 2^{\mathcal{P}}$  be the function such that  $\mathcal{D}(C) = \{P \mid C \text{ defends } P\}$ .

- $C$  is admissible if and only if  $C \subseteq \mathcal{D}(C)$ .
- $C$  is a complete extension if and only if  $C = \mathcal{D}(C)$ .
- $C$  is a grounded extension if and only if it is the smallest (with respect to set inclusion) complete extension.
- $C$  is a preferred extension if and only if it is a maximal (with respect to set inclusion) admissible extension.
- $C$  is a stable extension if and only if it is a preferred extension that attacks all arguments in  $\mathcal{P} \setminus C$ .

Dung's theory offers the choice among several alternatives to define when an individual argument is acceptable. First we have to choose a semantics, then we have to choose whether the argument must be in the union or intersection of the extensions of this semantics. We say that an argument is acceptable if it is in the union of all admissible sets (which is the same as being in the union of the complete extensions, or the union of the preferred extensions).

**Definition 8 (Acceptable proposal).** *A proposal is acceptable, if and only if it is in some admissible set.*

The following example illustrates that there may be multiple extensions.

*Example 5 (Continued from Example 4).* Reconsider the abstract dependence network visualised in Figure 4. Since there are four dependence relations, there are sixteen proposals.  $P_1$  and  $P_2$  attack each other, and each other proposal containing at most two relations attacks itself. Moreover,  $P_1$  and  $P_2$  attack all proposals containing three or four relations. Consequently, only the empty proposal,  $P_1$  and  $P_2$  are acceptable.

$P_1$  and  $P_2$  (Figure 4) consist of cycles, and do not contain multiple dependence relations for the same reason. In the remainder of this section, we show that this holds in general.

A proposal is exchange-based (or transaction-based) if every dependence is part of a cycle. This represents the fact that cooperation is based on reciprocity, called also *do-ut-des* [28].

**Definition 9 (Exchanged-based proposal).** *A cycle is a sequence of dependencies  $(s_1, t_1, d_1), \dots, (s_n, t_n, d_n)$  such that  $t_i = s_{i+1}$  for  $1 \leq i \leq n - 1$ , and  $t_n = s_1$ . A proposal  $P \subseteq D$  is exchange-based if and only if there is a set of cycles  $C$  such that  $P = \cup C$ .*

*Example 6 (Continued from Example 2).* Proposal  $P_1$  visualized in Figure 2.b is exchange-based, but proposal  $P_2$  visualized in Figure 2.c is not.

**Lemma 1.** *If  $P_1$  attacks  $P_2$  and  $P_1$  is a sub-proposal of  $P_2$ , then  $P_1$  is not acceptable.*

**Proposition 1.** *If a proposal is acceptable, then it is exchange-based.*

*Proof. (sketch) We prove the contrapositive. In an exchange-based proposal, each agent involved in a dependence performs something for another agent and depends on another agent. If a proposal is not exchange based, then there are two cases. First, it contains at least one agent who does not depend on another agent, but performs something for another agent. The payoff of this agent is negative, and the proposal therefore attacks itself. So it is not acceptable. Second, it contains an agent that depends on another agent, but does not perform something. This agent can be removed from the proposal, improving everyone else's payoff. So the proposal is attacked by a sub-proposal. It follows from Lemma 1 that the proposal is not part of an admissible set, and therefore it is not acceptable.*

If a reason occurs more than once in an abstract dependence network, it represents an OR-dependence. If it occurs more than once in a proposal, the proposal is redundant.

**Definition 10 (Non-redundant proposal).** *Proposal  $P \subseteq D$  is non-redundant if and only if each reason occurs at most once.*

**Proposition 2.** *If a proposal is acceptable, then it is non-redundant.*

*Proof. (sketch)* If a proposal is redundant, then it can be split in sub-proposals where one of the sub-proposals strictly dominates the original proposal for the agents involved in the sub-proposal. The result follows from Lemma 1.

**Proposition 3.** *If a proposal is exchange-based and non-redundant, then there are cost and benefit functions such that the proposal is acceptable.*

*Proof. (sketch)* Let there be  $n$  dependence relations in the abstract dependence network. Let the benefit of all dependencies be 1, let the cost of all dependencies in the proposal be  $1/(n + 1)$ , and the cost of all other dependencies be  $n$ . The payoff of the exchange-based proposal is positive for the agents involved in it.

### 3.3 Minimal proposals

In this section, we define a suitable notion of minimal proposal. For example, if proposals consist of sub-proposals of disconnected components, then the agents can negotiate the sub-proposals one at a time. However, the following example illustrates that we cannot restrict ourselves only to simple cycles.

*Example 7.* The benefit for agent  $a$  in Figure 2 of  $r_4$  or  $r_5$  individually may be smaller than the cost of  $r_1$ . For agent  $a$  only proposal  $P_1$  is acceptable, and  $P_2$  or  $P_3$  are not.

**Definition 11 (Minimal proposal).** *A proposal  $P$  is minimal if and only if it cannot be partitioned into two or more disjoint proposals  $P = P_1 \cup \dots \cup P_n$ , such that  $P$  is acceptable if and only if for  $1 \leq i \leq n$ :  $P_i$  is acceptable.*

The following proposition shows that if a proposal is minimal, then it cannot be split into two sub-proposals sharing at most one agent. Sauro [28] calls this the indecomposable do-ut-des property, abbreviated to i-dud.

**Proposition 4.** *If proposal  $P$  is minimal, then there are no disjoint nonempty proposals  $P_1$  and  $P_2$  such that  $P = P_1 \cup P_2$  and  $P_1$  and  $P_2$  share at most one agent.*

*Proof. (sketch)* We prove the contrapositive. If  $P_1$  or  $P_2$  is not acceptable, then  $P_1 \cup P_2$  is not acceptable. This follows from Lemma 1.

*Example 8.* Figure 5 visualizes an abstract dependence network and four acceptable proposals,  $P_1$  to  $P_4$ .  $P_3$  is a non-minimal proposal since it is acceptable only if  $P_1 \wedge P_2$  are acceptable. Proposal  $P_4$  attacks proposal  $P_3$ , and we have that the attack from  $P_4$  to  $P_3$  leads to an attack from  $P_4$  against  $P_2$ , which is a sub-proposal of  $P_3$ . Agent  $b$  is able to pay Amazon services to receive the book. The payment is made via PayPal. Dependence  $(p, l, r_7)$  may be read as “PayPal  $p$  depends on the linked debit card  $l$  to verify the account information” and  $(l, a, r_8)$  as “the linked debit card  $l$  depends on the Amazon services  $a$  to publish correct prices.”

**Proposition 5.** *If  $P$  is a proposal such that there are no disjoint nonempty proposals  $P_1$  and  $P_2$  such that  $P = P_1 \cup P_2$  and  $P_1$  and  $P_2$  share at most one agent, then there are cost and benefit functions such that  $P$  is minimal.*

*Proof. (sketch)* Let all costs be 1, and let all benefits be the in degree of the node plus 0.5.

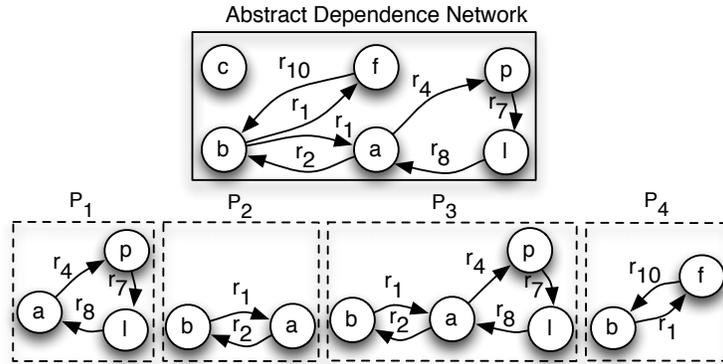


Fig. 5. The non-minimal proposal  $P_3$ .

## 4 Qualitative dependence theory

In this section we consider the case in which we do not know the cost and benefit functions, and we consider a more qualitative version of the dependence theory. We define preferred, complete and grounded proposals, not to be confused with preferred, complete and grounded sets of proposals defined in the argumentation theory.

### 4.1 $\exists$ -acceptable proposals

A proposal is called  $\exists$ -acceptable if it is acceptable for some cost and benefit functions, i.e. if it is exchange-based and non-redundant.

**Definition 12 ( $\exists$ -acceptable proposal).** A proposal  $P$  is  $\exists$ -acceptable if and only if it is non-redundant and exchange-based.

### 4.2 Preferred proposals

We define a *preferred proposal* as a proposal that obtains the maximal exchange. In other words, a preferred proposal is a maximal – with respect to set inclusion –  $\exists$ -acceptable proposal.

**Definition 13 (Preferred proposal).** A proposal  $P$  is preferred if and only if it is maximal (with respect to set inclusion) among the  $\exists$ -acceptable proposals.

The following example illustrates preferred proposals.

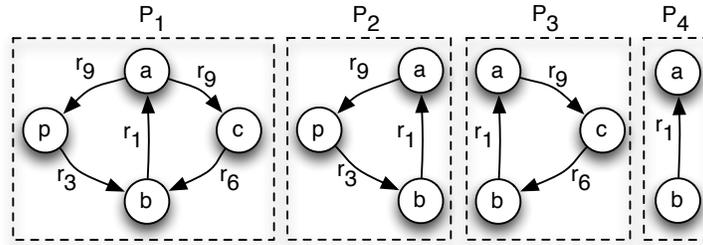
*Example 9 (Continued).* The dependence network visualized in Figure 3 contains three  $\exists$ -acceptable proposals,  $P_1, P_2$  and  $P_3$ . Only  $P_1$  is preferred.

It is straightforward to find preferred proposals of an abstract dependence network:

- While there are reasons occurring more than once in the proposal, remove one of them non-deterministically.
- While there are agents who perform but who do not depend on other agents, remove all dependence relations where these agents perform.

**Proposition 6.** *The intersection of preferred proposals does not have to be  $\exists$ -acceptable.*

*Proof.* Consider the example visualized in Figure 6. There are two preferred proposals  $P_2$  and  $P_3$ , and their intersection  $P_4$  is not exchange-based.



**Fig. 6.** Two preferred proposals  $P_2, P_3$  and intersection  $P_4$ .

**Proposition 7.** *There exists at least one preferred proposal.*

### 4.3 Complete, grounded and stable proposals

Based on the notion of redundancy, we have a notion of *conflict*: two  $\exists$ -acceptable proposals conflict if their union is not an  $\exists$ -acceptable proposal.

**Definition 14 (Uncontested proposal).** *Two  $\exists$ -acceptable proposals  $P_1$  and  $P_2$  conflict if  $P_1 \cup P_2$  is not  $\exists$ -acceptable. An uncontested proposal is a proposal which does not conflict with another proposal.*

**Proposition 8.** *Two  $\exists$ -acceptable proposals conflict if they contain two distinct dependences with the same reason.*

The following example illustrates a *contested* proposal, i.e., a proposal conflicting with another proposal.

*Example 10 (continued).* Consider the dependence network visualized in Figure 3. It contains two contested proposals  $P_1$  and  $P_2$ .

An uncontested proposal is a proposal which does not conflict with all the other proposals. Based on the notion of uncontested proposal, we define *complete* and *grounded* proposals.

**Definition 15 (Complete proposal).** *A complete proposal contains at least all the uncontested proposals.*

**Proposition 9.** *A preferred proposal is also a complete proposal.*

**Definition 16 (Grounded proposal).** *A grounded proposal contains precisely the uncontested proposals.*

*Example 11 (continued).* Consider again Figure 4. We have the following complete proposals:  $P_1$ ,  $P_2$ , and the empty proposal, i.e., if agents do not do anything. The only uncontested proposal is the empty set, because proposals  $P_1$  and  $P_2$  conflict. Therefore, the grounded proposal is the empty set.

**Proposition 10.** *There is a unique grounded proposal.*

**Definition 17 (Stable proposal).** *A stable proposal is a preferred proposal conflicting with all the other proposals.*

## 5 Related research

Emerson [18] was the first to introduce the theory of dependence in sociology, and Castelfranchi [13] popularized it in distributed artificial intelligence, and later in multi-agent systems, by exploiting the notion of social power.

In coalition formation, Sauro [27, 28] uses dependence networks to ensure that a coalition is effectively formed only when all its members agree on it, and they cannot deviate from what was established in the agreement. He introduces the do-ut-des and *indecomposable* do-ut-des property as primitives, whereas we derive it from the definition of minimality. While we use OR-dependence only, he uses also the notion of AND-dependence.

Other papers like the following couple coalition formation and the theory of dependence. Sichman [30], for instance, uses dependence networks to allow the agents to evaluate the susceptibility of other agents to adopt their goals. Grossi and Turrini [19] show how dependence-theoretic notions like cycles are amenable to a game-theoretic characterization. Finally, Bonzon *et al.* [9] and Sauro and Villata [29] use dependence networks in cooperative boolean games [17] to improve the computation of the pure-strategy Nash equilibrium and the core, respectively. Our work is orthogonal to this game theory research, as we consider arguing about agreements. In particular, we go beyond the use of dependence networks in coalition formation by introducing Dung's theory of abstract argumentation as reasoning technique to guide the agents in arguing during the process for reaching agreements.

Several papers propose to use argumentation theory to reason over the formation of coalitions of agents. Among them, Amgoud [1] uses a preference-based argumentation framework to represent coalition formation such that the preferred solutions to coalitional games are defined as preferred extensions of the corresponding argumentation framework. There are several differences with respect to this approach. In particular, the internal structure of coalitions is not specified in [1] as they are considered as abstract elements while here we represent their internal structure using dependence

networks which provide us a kind of explanation about *why* two coalitions cannot be formed at the same time going beyond preferences as in [1]. Bulling *et al.* [11] present a generalization of Dung’s theory, extended with a preference relation, such that ATL is used for reasoning about the behavior and abilities of the agents, and Cayrol and Lagasque-Schiex [15] define coalitions of arguments to reason over the acceptability of meta-arguments in the meta-level, and Bonzon *et al.* [8] translate argumentation frameworks into cooperative boolean games to compute preferred extensions using the pure-strategy Nash equilibrium. In this paper, we go beyond such a combination of argumentation theory and coalition formation techniques by introducing the theory of dependence which allows us to compute dependence-based agreements.

The work proposed by Boella *et al.* [7] combines argumentation theory, coalition formation, and dependence networks. They introduce a so called stability meta-argument which attacks one of these two attacks, preferring in this way one coalition over the other. A first difference is that they rely on dynamic dependence networks [12] where also higher-order dependencies are considered, while here we use standard dependence networks. The abstract theory proposed in this paper goes beyond the basic conflicts among coalitions considered by Boella *et al.* [7] introducing further constraints on the acceptability of proposals.

## 6 Summary and outlook

Our theory of arguing about dependence consists of two phases. In a first phase, proposals for agreement are selected from a dependence network. In a second phase, agents argue about the proposals using reasons—goals, tasks and resources—associated with the dependence relations to order the proposals. Dung’s semantics [16] determines acceptable proposals not dominated by other proposals.

The do-ut-des and indecomposable do-ut-des properties are derived from first principles, in the sense that argumentation provides the foundations for concepts such as redundancy, acceptability and minimality. We show that acceptable proposals are exchange based and non-redundant, and we show that minimal proposals cannot be split into nonempty sub-proposals sharing at most one agent. In addition, we show that if a proposal is exchange based and non-redundant, then there are cost and benefit functions such that the proposal is acceptable, and if a proposal cannot be split into several sub-proposals sharing at most one agent, then there are cost and benefit functions such that the proposal is minimal. This forms the basis for a more qualitative dependence theory based on  $\exists$ -acceptability.

One may wonder whether, instead of discarding them, it would not be useful to keep several redundant dependencies in the same proposal for the argumentation or negotiation process. However, as is shown in the argumentation phase, redundant dependence relations are represented by distinct proposals on which the agents argue. Thus, we do not discard them, both proposals for distinct performers will be used in the argumentation phase.

There are more insights from argumentation theory that can be used in the proposal phase. In particular, an important reason for the popularity of Dung’s abstract theory is that it can be applied to non-monotonic reasoning by instantiating the abstract argu-

ments with logical formulas [26], along the same lines as we have instantiated them with proposals for agreements. Moreover, Dung shows how his theory can be applied also to reasoning about games. Likewise, we can instantiate the abstract reasons with logical formulas representing goals, abilities, tasks and resources.

We made several assumptions in this paper to facilitate the presentation. Here we sketch their relaxation. First, we can introduce AND-dependences as defined by Sauro in [27], while retaining the results. Sauro defines cycles incorporating AND-dependence as going through one of the AND alternatives. Second, we can allow distinct performers for the same reason. To make the results hold again, we need to add additional assumptions. For example, if there are dependence relations with respect to the same reason with two distinct dependers and performers, then other dependence relations exist among them too: if  $(d_1, p_1, r), (d_2, p_2, r) \in D$ , then also  $(d_1, p_2, r), (d_2, p_1, r) \in D$ . Third, we can provide a cost-benefit analysis not depending on simple addition of costs and benefits. If we have a monotonicity property that more dependence increases benefits and more performance increases costs, then the results still hold.

We believe the combination of proposing agreements and arguing about proposals can be best understood from a dialogue perspective. Abstract argumentation theory as proposed by Dung is not very well suited for dialogue, since agents are abstracted away. It is more about inference from inconsistent knowledge bases than about merging knowledge bases [2], whereas dialogue is better seen as a generalization of merging knowledge bases. Moreover, most dialogue theories agree on the fact that arguments are proposed in some order. This seems a useful idea to incorporate in our theory. The question is, where does the order of proposing agreements come from? We may assume an ordering on dependencies (based on an ordering on the reasons) which gives us this ordering. Argumentation, and in particular negotiation, is based on preference (and indirectly, utility and probability). Finally, though we use argumentation, negotiation is an alternative model to come to an agreement [24].

## References

1. Amgoud, L.: An argumentation-based model for reasoning about coalition structures. In: Proc. of ArgMAS, LNCS 4049. pp. 217–228. Springer (2005)
2. Amgoud, L., Kaci, S.: An argumentation framework for merging conflicting knowledge bases. *Int. J. Approx. Reasoning* 45(2), 321–340 (2007)
3. Amgoud, L., Prade, H.: Reaching agreement through argumentation: A possibilistic approach. In: Proc. of KR. pp. 175–182. AAAI Press (2004)
4. Billhardt, H., Centeno, R., Cuesta, C.E., Fernández, A., Hermoso, R., Ortiz, R., Ossowski, S., Pérez-Sotelo, J.S., Vasirani, M.: Organisational structures in next-generation distributed systems: Towards a technology of agreement. *Multiagent and Grid Systems* 7(2-3), 109–125 (2011)
5. Boella, G., Sauro, L., van der Torre, L.W.N.: An abstraction from power to coalition structures. In: ECAI (2004)
6. Boella, G., Sauro, L., van der Torre, L.W.N.: From social power to social importance. *Web Intelligence and Agent Systems* 5(4), 393–404 (2007)
7. Boella, G., van der Torre, L., Villata, S.: Analyzing cooperation in iterative social network design. *J. UCS* 15(13), 2676–2700 (2009)

8. Bonzon, E., Devred, C., Lagasquie-Schiex, M.C.: Argumentation and cp-boolean games. *International Journal on Artificial Intelligence Tools* 19(4), 487–510 (2010)
9. Bonzon, E., Lagasquie-Schiex, M.C., Lang, J.: Dependencies between players in boolean games. *Int. J. Approx. Reasoning* 50(6), 899–914 (2009)
10. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems* 8, 203–236 (2004)
11. Bulling, N., Dix, J., Chesñevar, C.I.: Modelling coalitions: Atl + argumentation. In: *Proc. of AAMAS*. pp. 681–688. IFAAMAS (2008)
12. Caire, P., Villata, S., Boella, G., van der Torre, L.: Conviviality masks in multiagent systems. In: *Proc. of AAMAS*. pp. 1265–1268. IFAAMAS (2008)
13. Castelfranchi, C.: The micro-macro constitution of power. *Protosociology* 18, 208–269 (2003)
14. Castelfranchi, C., Cesta, A., Conte, R., Miceli, M.: Foundations for Interaction: The Dependency Theory. In: *Proc. of AI\*IA, LNCS 728*. pp. 59–64. Springer (1993)
15. Cayrol, C., Lagasquie-Schiex, M.C.: Coalitions of arguments: A tool for handling bipolar argumentation frameworks. *Int. J. Intell. Syst.* 25(1), 83–109 (2010)
16. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.* 77(2), 321–358 (1995)
17. Dunne, P.E., van der Hoek, W., Kraus, S., Wooldridge, M.: Cooperative boolean games. In: *Proc. of AAMAS*. pp. 1015–1022. IFAAMAS (2008)
18. Emerson, R.: Power-dependence relations. *American Sociological Review* 27, 31–41 (1962)
19. Grossi, D., Turrini, P.: Dependence in games and dependence games. *Autonomous Agents and Multi-Agent Systems* 25(2), 284–312 (2012)
20. Kraus, S., Sycara, K.P., Evenchik, A.: Reaching agreements through argumentation: A logical model and implementation. *Artif. Intell.* 104(1-2), 1–69 (1998)
21. Malone, T., Crowston, K.: The interdisciplinary study of coordination. *Computing Surveys* 26 (1), 87119 (1994)
22. Malone, T.W., Crowston, K.: The interdisciplinary study of coordination. *ACM Comput. Surv.* 26(1), 87–119 (1994)
23. von Neumann, J., Morgenstern, O.: *Theory of Games and Economic Behaviour*. Princeton University Press (1944)
24. Rahwan, I., Pasquier, P., Sonenberg, L., Dignum, F.: On the benefits of exploiting underlying goals in argument-based negotiation. In: *Proc. of AAI*. pp. 116–121. AAAI Press (2007)
25. Rahwan, I., Ramchurn, S.D., Jennings, N.R., Mcburney, P., Parsons, S., Sonenberg, L.: Argumentation-based negotiation. *Knowl. Eng. Rev.* 18(4), 343–375 (2003)
26. Rahwan, I., Simari, G. (eds.): *Argumentation in Artificial Intelligence*. Springer (2009)
27. Sauro, L.: Formalizing Admissibility Criteria in Coalition Formation among Goal Directed Agents. Ph.D. thesis, University of Torino (2006)
28. Sauro, L.: Qualitative criteria of admissibility for enforced agreements. *CMOT* 12(2-3), 147–168 (2006)
29. Sauro, L., Villata, S.: Dependency in cooperative boolean games. *J Logic Comp.*, 10.1093/logcom/exr030 (2011)
30. Sichman, J.S.: Depint: Dependence-based coalition formation in an open multi-agent scenario. *J of Artificial Societies and Social Simulation* 1(2) (1998)
31. Sichman, J.S., Conte, R.: Multi-agent dependence by dependence graphs. In: *Proc. of AAMAS*. pp. 483–490. IFAAMAS (2002)
32. Visser, W., Hindriks, K.V., Jonker, C.M.: Interest-based preference reasoning. In: *Proc. of ICAART*. pp. 79–88. SciTePress (2011)
33. Yu, E.: Modelling Strategic Relationships for Process Reengineering. Ph.D. thesis, University of Toronto (1995)