Conditional Acceptance Functions

Richard Booth, Souhila Kaci, Tjitze Rienstra, Leendert Van Der Torre

Conditional Acceptance Functions

Richard BOOTH [a] Souhila KACI [b] Tjitze RIENSTRA [a] and
Leendert VAN DER TORRE [a]
[a] *University of Luxembourg*
[b] *LIRMM, University of Montpellier 2, France*

Abstract. Dung-style abstract argumentation theory centers on argumentation frameworks and acceptance functions. The latter take as input a framework and return sets of labelings. This methodology assumes full awareness of the arguments relevant to the evaluation. There are two reasons why this is not satisfactory. Firstly, full awareness is, in general, not a realistic assumption. Second, frameworks have explanatory power, which allows us to reason abductively or counterfactually, but this is lost under the usual semantics. To recover this aspect, we generalize conventional acceptance, and we present the concept of a conditional acceptance function.

Keywords. Abstract argumentation, Semantics, Dynamics

1. Introduction

Dung-style abstract argumentation theory [8] centers on two concepts. The first is that of an *argumentation framework* (in short, framework). This is a pair consisting of a set of arguments and an attack relation. The arguments are modeled as abstract entities, and the attack relation encodes conflicts between arguments. The second concept is that of an *acceptance function*. It takes as input a framework and returns a set of extensions or labelings, that represent rational evaluations of the arguments, under some set of criteria. These criteria are embodied in what is called a semantics. Examples usually considered are complete, grounded, preferred and stable semantics, where each semantics defines a corresponding acceptance function.

This methodology assumes full awareness of the arguments relevant to the evaluation. This is reflected by the notions of rationality employed by the usual semantics, where the status of an argument is determined under the assumption that all its attackers are known. For example, an argument that is not attacked, or that is never attacked by an accepted argument, is *always* accepted.

There are two reasons why this is not satisfactory. Firstly, argumentation is an inherently dynamic process. In recent years, dynamics aspects of argumentation have been investigated in the literature. However, research has focused mainly on the effect of changing a framework (e.g. [3,4,5,7,10,11]). But more generally, a realistic agent does not reason under the assumption of complete knowledge, and it may be necessary to consider the possibility that new arguments come into play during the evaluation of the framework. Second, and more to the point, frameworks encode more information than what the labelings of the framework reveal, under the usual semantics. Consider, for example, frameworks (a)...(e), shown in figure 1. They all have the same unique complete labeling, namely the one in which the arguments a and c are accepted and b is rejected. Thus, the frameworks are indistinguishable for complete, grounded, preferred and stable semantics. However, while all frameworks give rise to the same evaluation, we may say that they all encode a different explanation for this evaluation. It is exactly this *explanatory* aspect of argumentation that we lose under the usual semantics. This aspect allows us to reason abductively or counterfactually (What if a was not accepted? For what reason could b be rejected? Etcetera.) To recover this aspect, we need to generalize the conventional concept of argument acceptance.

This is what we will do in this paper; we present the concept of a *conditional acceptance function*, which provides a conceptually richer account of argument evaluation. It generalizes conventional acceptance in a simple way: instead of taking only a framework as input, it additionally takes as input a *condition*. The labelings returned are the labelings of the framework *given* the condition. The condition takes the form of a set of arbitrary labelings, from which the conditional acceptance function selects those that are, according to the type of the function, considered 'most rational'.

The plan of this paper is as follows. In section 3, we present the basic concept of a conditional acceptance function. In section 4, we present an instance of this concept: *conditional completeness*. In section 5 we conclude and suggest some directions for future work. Before we start, however, we recall in the following section the basic definitions of Dung-style argumentation with labeling based semantics.

2. Formal preliminaries

The central notion in argumentation theory is that of an argumentation framework:

Definition 2.1. *An* argumentation framework F *is a pair* (A_F, R_F)*, where* A_F *is a finite set of* arguments*, and* $R_F \subseteq A_F \times A_F$ *is the* attack relation*.*

We take the labeling-based approach [1,6] rather than the extension-based approach. An evaluation of the arguments of a framework is then represented by a *labeling* that assigns to each argument a label I, O or U (that is, *in*, *out* or *undecided*, where in and out are also called *accepted* and *rejected*).

Definition 2.2. *Given a framework* F*, a* labeling *is a function* $L : A_F \to V$*, where* $V = \{I, U, O\}$*. We denote the set of all labelings of* F *by* $\mathbb{L}_F^{all}$*. If* L *is a labeling of* F *then* $L^{-1} : V \to \mathscr{P}(A_F)$*, where* $\mathscr{P}$ *denotes the power set operation, is defined by* $L^{-1}(v) = \{a \in A_F \mid L(a) = v\}$*.*

Given a framework, an acceptance function returns the set of 'best' labelings, each of which represents a rational evaluation of the arguments.

Definition 2.3. *An* acceptance function *is a function* $\mathbb{A}$ *that returns, for any framework* F*, a set* $\mathbb{A}_F \subseteq \mathbb{L}_F^{all}$*.*

For consistency with later sections, we write the argument F as a subscript, i.e. $\mathbb{A}_F$. Different types of acceptance functions can be defined, each corresponding to an argumentation semantics that embodies a particular set of rationality criteria. The properties of *conflict-freeness* and *admissibility* are usually considered to be the minimum re-

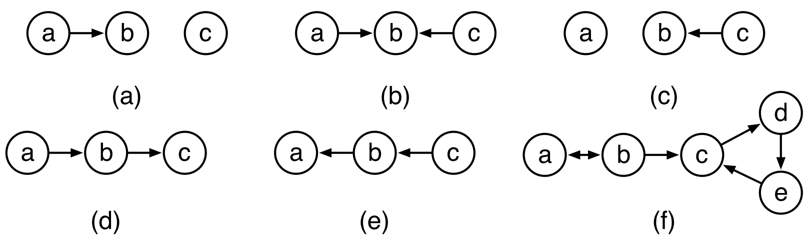

Figure 1. Six abstract frameworks.

quirements. Conflict-freeness states that simultaneously accepted arguments cannot attack each other. Admissibility states that every accepted argument is defended by the set of accepted arguments. Formally:

Definition 2.4. *Given a framework F, we say that a labeling $L \in \mathbb{L}_F^{all}$ is* conflict-free[1] *iff: $\nexists(a,b) \in R_F$ s.t. $L(a) = L(b) = I$; and* admissible *iff: $\forall a \in A_F$, $L(a) = I$ implies $\forall(b,a) \in R_F, L(b) = O$ and $L(a) = O$ implies $\exists(b,a) \in R_F$ s.t. $L(b) = I$. We denote the set of conflict-free and admissible labelings of F by $\mathbb{L}_F^{cf}$ and $\mathbb{L}_F^{ad}$, respectively.*

Another property is *completeness*. It essentially captures the assumption of full awareness of the arguments relevant to the evaluation. More precisely, in a complete labeling, any argument that *can* be labeled *I*, or otherwise *U*, without violating admissibility, *is* labeled *I*, or otherwise *U*.

Definition 2.5. *Given a framework F, we say that a labeling L is* complete *iff $\forall a \in A_F$, $L(a) = I$ iff $\forall(b,a) \in R_F, L(b) = O$ and $L(a) = O$ iff $\exists(b,a) \in R_F, L(b) = I$.*

Among the complete labelings, we can select those that satisfy certain properties, such as having a maximal or minimal set of arguments labeled *I*, or having no argument labeled *U*. These give rise to the *preferred*, *grounded* and *stable* semantics. The acceptance functions corresponding to the complete, grounded, preferred and stable semantics, are defined below.

Definition 2.6. *The* complete acceptance function*, denoted by $\mathbb{A}^{co}$, is defined to return all complete labelings of F. The* preferred, grounded *and* stable *acceptance functions, denoted by $\mathbb{A}^{pr}, \mathbb{A}^{gr}$ and $\mathbb{A}^{st}$, respectively, are defined as follows.*

- $\mathbb{A}_F^{pr} = \{L \in \mathbb{A}_F^{co} \mid \nexists K \in \mathbb{A}_F^{co}, K^{-1}(I) \supset L^{-1}(I)\}$
- $\mathbb{A}_F^{gr} = \{L \in \mathbb{A}_F^{co} \mid \nexists K \in \mathbb{A}_F^{co}, K^{-1}(U) \supset L^{-1}(U)\}$
- $\mathbb{A}_F^{st} = \{L \in \mathbb{A}_F^{co} \mid L^{-1}(U) = \emptyset\}$

Example 2.7. *Consider framework (f) in figure 1. There are three complete labelings, namely $(IOUUU), (OIOIO)$ and $(UUUUU)$[2]. The labeling $(UUUUU)$ is also the*

[1] Different definitions of conflict-free labelings exist. In [1], a conflict-free labeling is defined by the condition that, $\forall a \in A_F$, $L(a) = I$ implies $\nexists(b,a) \in R_F$ s.t. $L(b) = I$, and $L(a) = O$ implies $\exists(b,a) \in R_F$ s.t. $L(a) = I$. Our definition is the same as in [6].

[2] We will denote labelings by sequences of the form $(v_a v_b \ldots)$, meaning $L(a) = v_a, L(b) = v_b$ etc., where it is clear from context what framework we refer to, and what the arguments $a, b, \ldots$ are.

grounded labeling. The labelings $(IOUUU)$ *and* $(OIOIO)$ *are preferred labelings. Only* $(OIOIO)$ *is a stable labeling.*

3. Conditional acceptance functions

The concept of an acceptance function, as introduced above, is straightforward; it takes as input a framework, and returns a set of labelings. In this section we introduce the concept of a *conditional acceptance function*. Intuitively, it represents a method of argument evaluation where the agent has a set of labelings that satisfy some prior conditions, and he is interested in finding those that are, in the same sense as before, rational.

Definition 3.1. *A* conditional acceptance function *is a function* $\mathbb{CA}_F : \mathscr{P}(\mathbb{L}_F^{all}) \to \mathscr{P}(\mathbb{L}_F^{all})$ *such that* $\mathbb{CA}_F(X) \subseteq X$*. When* $L \in \mathbb{CA}_F(X)$*, we say that* L *is a labeling* conditional on X.

When we use a conditional acceptance function, we have control over which labelings we consider to be evaluated. This set may be restricted to labelings satisfying some constraint. The constraint is the condition under which the framework is evaluated. A conditional acceptance function may *generalize* an acceptance function. Formally:

Definition 3.2. *A conditional acceptance function* $\mathbb{CA}_F$ generalizes *an acceptance function* $\mathbb{A}_F$ *only if* $\mathbb{CA}_F(\mathbb{L}_F^{all}) = \mathbb{A}_F$.

In words, a conditional acceptance function generalizes an acceptance function only if it coincides in the case where the input is not constrained and consists of all labelings.

4. Conditional completeness

In this section we present an instance of a conditional acceptance function: the *conditionally complete* acceptance function. In the introduction we pointed out that agents, when they evaluate a framework, need to consider the possibility that new arguments come into play. Basically, this means that the assumption of full awareness must be dropped, and arguments are not only labeled O or U if they have an I or U labeled attacker, but also when this is enforced by the condition. So, a conditionally complete labeling is not necessarily complete. We will first answer the question: what are the minimal criteria satisfied by a conditionally complete labeling? We turn to this in the following section.

4.1. Subcomplete labelings

A conditional labeling will, in general, not be admissible, because we may have a condition saying that some argument a is rejected, i.e., $L(a) = O$ for all labelings in the conditioning set X. Then, the arguments attacked by a need not be defended from a by an accepted argument. Rather, it is defended by the condition. A weaker property, that will turn out to be adequate, is what we will call subcompleteness:

Definition 4.1. *Given a framework F, we say that a labeling L is* subcomplete *iff: if $\forall a \in A$, if $L(a) = I$ then for every neighbor b of a, $L(b) = O$, where a neighbor of a is an argument b such that $(a,b) \in R_F$ or $(b,a) \in R_F$. We denote the set of subcomplete labelings by $\mathbb{L}^{sc}_F$.*

Note that subcompleteness is stronger than conflict-freeness, which does not exclude cases where $(a,b) \in R_F$ and $L(a) = I$ and $L(b) = U$ or vice versa. However, it is, like conflict-freeness, a symmetric property, in that it does not depend on the direction of the attack relation. We motivate subcompleteness by pointing out that it is characterized by another property, which we call *embeddability*. Intuitively, every subcomplete labeling of a framework F is 'embedded' in a complete labeling of a framework that extends F with additional arguments and attacks. Conversely, every labeling that is embedded in a complete labeling, is subcomplete. To formalize this, we first need to introduce the concept of *expansion* and *restriction*.

Definition 4.2. *Let F,G be two frameworks. We say that G is an* expansion *of F if and only if $A_F \subseteq A_G$ and $R_F \subseteq R_G$.*

Definition 4.3. *Given a framework F and a set $B \subseteq A_F$, the B-*restriction of a labeling *$L \in \mathbb{L}^{all}_F$, is a function $L{\downarrow}B : B \to V$ defined by $L{\downarrow}B(a) = L(a)$. The B-*restriction of a set *$X \subseteq \mathbb{L}^{all}_F$, denoted by $X{\downarrow}B$, is defined by $X{\downarrow}B = \{L{\downarrow}B \mid L \in X\}$. The B-*restriction of F, *denoted by $F{\downarrow}B$ is defined by $F{\downarrow}B = (B, R_F \cap B \times B)$.*

Formally, embeddability is defined as follows.

Definition 4.4. *Given a framework F, we say that a labeling $L \in \mathbb{A}_F$ is* embeddable *if and only if there is an expansion G of F such that $\exists K \in \mathbb{A}^{co}_G$ s.t. $K{\downarrow}A_F = L$.*

Proposition 4.5. *A labeling L is embeddable if and only if L is subcomplete labeling.*

In the following section, we define the conditionally complete acceptance function $\mathbb{CA}^{co}_F$. The property of subcompleteness will be a minimal condition for a labeling to be considered conditionally complete, i.e. $\mathbb{CA}^{co}_F$ will return only subcomplete labelings.

4.2. The conditionally complete acceptance function

Now, how do we determine, given a set of subcomplete labelings, what the set of 'best' subcomplete labelings is? Our aim is to follow the intuition behind complete semantics. This means that 'best' should be interpreted as 'most complete'.

Recall how completeness is defined (definition 2.5). We can say that a complete labeling assigns to each argument the strongest label possible, given its attackers. This notion of strength is captured by the following order:

Definition 4.6. *The order $\geq$ is a transitive reflexive order over $\{I,U,O\}$ such that $I > U > O$, where $>$ is defined as usual. The order $\succeq$ is an ordering over labelings defined as follows: $L \succeq K$ iff, $\forall a \in A, L(a) \geq K(a)$, where $\succ$ is defined as usual. If $v > w$ we say that v is* stronger *than w. Similarly, if $L \succ K$ we say that L is stronger than K.*

X	$\mathbb{CA}_F^{co}(X)$
$\{(OI),(IO),(UU),(UO),(OU),(OO)\}$	$\{(OI),(IO),(UU)\}$
$\{(IO),(UO),(OO)\}$	$\{(IO)\}$
$\{(UU),(UO),(OU),(OO)\}$	$\{(UU)\}$
$\{(UO),(OO)\}$	$\{(UO)\}$
$\{(OO)\}$	$\{(OO)\}$

Table 1. Some results of $\mathbb{CA}_F^{co}$ for the framework of example 4.9

If we choose among an arbitrary set X of subcomplete labelings, then it is reasonable to say that, if there is no labeling in X that assigns I to an argument a, then the strongest label we can assign to a is U. Similarly, if all labelings in X assign O to a, then O is the strongest label we can assign to a. Following this intuition, we define the concept of *completeness given* X, of a labeling $L \in X$, as follows.

Definition 4.7. *Given a framework F and a set $X \subseteq \mathbb{L}_F^{sc}$, we say that a labeling $L \in X$ is* complete given X *iff $\forall a \in A_F$:*

1. *If $L(a) = U$ then either $(\forall K \in X, K(a) \leq U)$ or $\exists (b,a) \in R_F, L(b) = U$.*
2. *If $L(a) = O$ then either $(\forall K \in X, K(a) = O)$ or $\exists (b,a) \in R_F, L(b) = I$.*

Definitions 4.1 and 4.7 imply that a labeling L is complete given the set of *all* subcomplete labelings if and only if L is complete. The conditionally complete acceptance function, which generalizes the complete semantics, is now defined to return all subcomplete labelings that are complete, given the set of subcomplete labelings in the input.

Definition 4.8. *Given a framework F, the* conditionally complete acceptance function $\mathbb{CA}_F^{co}$ *is a conditional acceptance function defined by $\mathbb{CA}_F^{co}(X) = \{L \in X \cap \mathbb{L}_F^{sc} \mid L$ is complete given $X \cap \mathbb{L}_F^{sc}\}$.*

Example 4.9. *Let F be the framework (A_F, R_F) with $A_F = \{a,b\}$ and $R_F = \{(a,b),(b,a)\}$. The subcomplete labelings of F are $(IO),(OI),(UU),(UO),(OU)$ and (OO). Table 1 shows the results of $\mathbb{CA}_F^{co}$ for some sets of subcomplete labelings of F.*

A problem is that, according to definition 4.7, not every set X has a labeling that is complete given X. Consider, for example, a framework consisting of a cycle of three arguments, and a set $X = \{(IOO),(OIO),(OOI)\}$. We then have that $\mathbb{CA}_F^{co}(X) = \emptyset$. So, not every possible set of subcomplete labelings has a 'most rational' one, according to $\mathbb{CA}_F^{co}$ and this may be undesirable. However, we do have that $\mathbb{CA}_F^{co}$ is nonempty, if the input is *defined by an upper bound*. The idea, here, is that we can restrict the strength of the labels of certain arguments by an upper bound. We can express this by a labeling L, where each label is an upper bound on the label (i.e. the strongest label) that may be assigned to the argument. If a set X is a maximal set of subcomplete labelings that all satisfy an upper bound L, we will say that X is *defined by the upper bound* L:

Definition 4.10. *Given a framework F, and labeling $L \in \mathbb{L}_F^{all}$, we say that a set X is* defined by the upper bound L *if $X = \{K \in \mathbb{L}_F^{sc} \mid K \preceq L\}$. We denote the set defined by L by $[L]_F$.*

Example 4.11. *Let F be the same framework as in example 4.9. The set of subcomplete labelings of F is $\mathbb{L}_F^{sc} = \{(IO),(OI),(UU),(UO),(OU),(OO)\}$. Suppose we condition* *on b being labeled O. This is expressed as an upper bound* (IO)*, resulting in the set* $[(IO)]_F = \{(IO), (UO), (OO)\}$*. Now suppose a cannot be labeled I. This is expressed as an upper bound* (UI)*, resulting in* $[(UI)]_F = \{(OI), (UU), (UO), (OU), (OO)\}$*. Note that, in these examples, we have a subcomplete labeling that is complete. This is, in general, not the case. Consider, for example, the upper bound* (OU)*. We then get* $[(OU)]_F = \{(OU), (OO)\}$*. Finally, some examples of sets that are not defined by an upper bound are* $\{(OI)\}$*,* $\{(IO), (UU)\}$ *and* $\{(UU), (OO)\}$*.*

We now have the following result.

Proposition 4.12. $\forall L \in \mathbb{L}_F^{all}$, $\mathbb{CA}_F^{co}([L]_F) \neq \emptyset$.

Example 4.13. *Consider framework (f) shown in figure 1. The complete labelings are* $(IOUUU), (OIOIO)$ *and* $(UUUUU)$*. The complete labelings, given that d is labeled out, are* $\mathbb{CA}_F([(IIIOI)]_F) = \{(IOOOI), (OIOOI), (UUOOI)\}$*.*

4.3. Conditional directionality

An important property of a semantics is *directionality* [1,2], which was shown to be satisfied by the complete, grounded and preferred acceptance functions. It states that the label of an argument is determined only by the labels of its attackers (and indirectly, those of the attackers of the attackers, etc.). To formalize this property, we need to introduce the concept of *unattacked set*.

Definition 4.14. *Given a framework F, we call a set* $B \subseteq A_F$ *an* unattacked set *iff* $\forall a \in A_F \setminus B$*,* $\nexists b \in B$ *s.t.* $(a,b) \in R_F$*. We denote the set of unattacked sets of F by* $U(F)$*.*

Directionality: If $X \in U(F)$ then $\mathbb{A}_F{\downarrow}X = \mathbb{A}_{F\downarrow X}(L{\downarrow}X)$.

We present here the conditional version of this property. It states that the label of an argument depends on restrictions imposed on its (indirect) attackers, and not on restrictions imposed on other arguments. Formally:

Conditional directionality: $\forall X \in U(F)$, if $L{\downarrow}X = K{\downarrow}X$ then $\mathbb{CA}_F([L]_F){\downarrow}X = \mathbb{CA}_F([K]_F){\downarrow}X$.

Proposition 4.15. $\mathbb{CA}_F^{co}$ *satisfies conditional directionality.*

Example 4.16. *Let F be the framework shown in figure 1(d). One unattacked set is* $\{a, b\}$*. According to conditional directionality, the labels of this set should not be affected when we restrict the label of c. Suppose we restrict c to O, i.e. we impose the upper bound* (IIO)*. One labeling satisfying this upper bound is* (OIO)*, but this result would violate directionality, as it labels b I, while b is not (indirectly) attacked by c. Instead we have* $\mathbb{CA}_F^{co}([(IIO)]_F) = \{(IOO)\}$*.*

5. Conclusions and further work

In this paper we have introduced a generalization of the concept of an acceptance function, in the form of a conditional acceptance function. These concepts provide a conceptually richer account of argument evaluation, that accounts for dynamic aspects of argument evaluation, and that recovers the explanatory aspects of frameworks. We have shown the use of conditional acceptance by a number of examples, and we have proved that our generalization satisfies a conditional version of directionality.

One direction for future work is to apply the generalization to other semantics. Furthermore, we are interested in properties that characterize the behavior of conditional acceptance functions. In this paper, we have considered only one (conditional directionality). We may benefit from a link with the study of choice functions, as studied in economics [9]. Also, the explanatory power of abstract argumentation seems interesting to explore, and this work can be seen as a first step.

Finally, we believe that the ideas described in this paper have many potential applications. The problem that argument evaluation is a dynamic process, that may depend on external factors, appears in many areas where argumentation theory is applied. A final direction for future work is to generalize the dialogical interpretation of argumentation. Work in this area has mainly focused on dialogical proof theories, that assume a fixed framework, shared between both the opponent and proponent. It is interesting to model a more realistic setting, where agents have beliefs about the knowledge and goals of other agents, and need to adapt their strategies accordingly. Some work in this direction is [12,13]. We plan to approach this problem by building on the ideas in the current paper.

References

[1] P. Baroni, M.W.A Caminada, and M. Giacomin. An introduction to argumentation semantics. *Knowledge Eng. Review*, 26(4):365–410, 2011.

[2] P Baroni and M Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence*, 171:675 – 700, 2007.

[3] P. Bisquert, C. Cayrol, F.D. de Saint-Cyr, and M.C. Lagasquie-Schiex. Duality between addition and removal: a tool for studying change in argumentation. *IPMU 2012*.

[4] G. Boella, S. Kaci, and L. van der Torre. Dynamics in argumentation with single extensions: Abstraction principles and the grounded extension. In Claudio Sossai and Gaetano Chemello, editors, *ECSQARU*, volume 5590 of *Lecture Notes in Computer Science*, pages 107–118. Springer, 2009.

[5] G. Boella, S. Kaci, and L. van der Torre. Dynamics in argumentation with single extensions: Attack refinement and the grounded extension (extended version). In Peter McBurney, Iyad Rahwan, Simon Parsons, and Nicolas Maudet, editors, *ArgMAS*, volume 6057 of *Lecture Notes in Computer Science*, pages 150–159. Springer, 2009.

[6] M.W.A. Caminada and D.M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009.

[7] C. Cayrol, F.D. de Saint-Cyr, and M.C. Lagasquie-Schiex. Change in abstract argumentation frameworks: Adding an argument. *Journal of Artificial Intelligence Research*, 38(1):49–84, 2010.

[8] P.M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.

[9] S.O. Hansson and T Grüne-Yanoff. Preferences. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2011 edition, 2011.

[10] B. Liao, L. Jin, and R.C. Koons. Dynamics of argumentation systems: A division-based method. *Artificial Intelligence*, 2011.

[11] M.O. Moguillansky, N.D. Rotstein, M.A. Falappa, A.J. García, and G.R. Simari. Argument theory change through defeater activation. *Proc. of COMMA 2010*, pages 359–366, 2010.

[12] N. Oren and T. Norman. Arguing using opponent models. *Argumentation in Multi-Agent Systems*, pages 160–174, 2010.

[13] M. Thimm and A. García. On strategic argument selection in structured argumentation systems. *Argumentation in Multi-Agent Systems*, pages 286–305, 2011.