On the Functional Completeness of Argumentation Semantics*

Massimiliano Giacomin[1], Thomas Linsbichler[2], and Stefan Woltran[2]

[1] University of Brescia, Italy
[2] TU Wien, Austria

Abstract. Abstract argumentation frameworks (AFs) are one of the central formalisms in AI; equipped with a wide range of semantics, they have proven useful in several application domains. We contribute to the systematic analysis of semantics for AFs by connecting two recent lines of research – the work on input/output frameworks and the study of the expressiveness of semantics. We do so by considering the following question: given a function describing an input/output behaviour by mapping extensions (resp. labellings) to sets of extensions (resp. labellings), is there an AF with designated input and output arguments realizing this function under a given semantics? For the major semantics we give exact characterizations of the functions which are realizable in this manner.

1 Introduction

Dung's argumentation frameworks (AFs) have been extensively investigated, mainly because they represent an abstract model unifying a large variety of specific formalisms ranging from nonmonotonic reasoning to logic programming and game theory [9]. After the development and analysis of different semantics [13, 6, 2], recent attention has been drawn to their expressive power, i.e. determining which sets of extensions [10] and labellings [11] can be enforced in a single AF under a given semantics. Such results have recently been facilitated in order to express AGM-based revision in the context of abstract argumentation [8].

In [1], it has been shown that an AF can be viewed as a set of partial interacting sub-frameworks each characterized by an input/output behavior, i.e. a semantics-dependent function which maps each labelling of the "input" arguments (the external arguments affecting the sub-framework) into the set of labellings prescribed for the "output" arguments (the arguments of the sub-framework affecting the external ones). It turns out that under the major semantics, i.e. complete, grounded, stable and (under some mild conditions) preferred semantics, sub-frameworks with the same input/output behavior can be safely exchanged, i.e. replacing a sub-framework with an equivalent one does not affect the justification status of the invariant arguments: semantics of this kind are called *transparent* [1].

* This research has been supported by the Austrian Science Fund (FWF) through projects I1102 and P25521.

Then, somewhat resembling functional completeness of a specific set of logic gates, a natural question concerns the expressive power of transparent semantics in the context of an interacting sub-framework: given a so-called *I/O specification*, i.e. a function describing an input/output behaviour by mapping extensions (resp. labellings) to sets of extensions (resp. labellings), is there an AF with designated input and output arguments realizing this function under a given semantics? In this paper, we answer this question as follows:

- For the stable, preferred, semi-stable, stage, complete, ideal, and grounded semantics we exactly characterize all realizable two-valued I/O specifications.
- For the preferred and grounded labellings we exactly characterize all realizable three-valued I/O specifications.

Answering this question is essential in many aspects. First, it adds to the analysis and *comparison of semantics* (see e.g. [5, 3]), by providing an absolute characterization of their functional expressiveness, which holds independently of how the abstract argumentation framework is instantiated. Second, it lays foundations towards a theory of *dynamic and modular argumentation*. More specifically, a functional characterization provides a common ground for different representations of the same sub-framework, e.g. to devise a summarized version of a sub-framework, or to give an argumentation-based view of the same framework at different levels of abstraction, as in metalevel argumentation [12]. One may also translate a different formalism to an AF or vice versa, or provide an argument-free representation of a given AF for human/computer interaction issues: in all of these cases, it is important to know whether an input/output behavior is realizable under a given argumentation semantics. Finally, our results are important in the dynamic setting of *strategic argumentation*, where a player may exploit the fact that for some set of arguments certain labellings are achievable (or non achievable) independently of the labelling of other arguments, or more generally she/he may exploit knowledge on the set of realizable dependencies. For example, an agent may desire to achieve some goal, i.e., ensure that a certain argument is justified. Considering arguments brought up by other agents as input arguments, our results enable to verify whether the goal is achievable and provide one particular way for the agent to bring up further arguments in order to succeed.

The paper is organized as follows. After providing the necessary background in Section 2, Section 3 introduces the notion of *I/O*-gadget to represent a sub-framework, and tackles the above problem with extension-based two-valued specifications. Labelling-based three-valued specifications are investigated in Section 4 and Section 5 concludes the paper.

2 Background

We assume a countably infinite domain of arguments $\mathfrak{A}$. An *argumentation framework* (AF) is a pair $F = (A, R)$ where $A \subseteq \mathfrak{A}$ and $R \subseteq A \times A$. We assume that A is non-empty and finite. For an AF $F = (A, R)$ and a set of arguments

$S \subseteq A$, we define $S^+_F = \{a \in A \mid \exists s \in S : (s,a) \in R\}$, $S^\oplus_F = S \cup S^+_F$, and $S^-_F = \{a \in A \mid \exists s \in S : (a,s) \in R\}$.

Given $F = (A,R)$, a set $S \subseteq A$ is *conflict-free* (in F), if there are no arguments $a,b \in S : (a,b) \in R$. An argument $a \in A$ is *defended* (in F) by a set $S \subseteq A$ if $\forall b \in A : (b,a) \in R \Rightarrow b \in S^+_F$. A set $S \subseteq A$ is *admissible* (in F) if it is conflict-free and defends all of its elements. We denote the set of conflict-free and admissible sets in F as $cf(F)$ and $ad(F)$, respectively.

An extension-based semantics σ associates to any $F = (A,R)$ the (possibly empty) set $\sigma(F)$ including subsets of A called σ-extensions. In this paper we focus on complete, grounded, preferred, ideal, stable, stage and semi-stable semantics, with extensions defined as follows:

- $S \in co(F)$ iff $S \in ad(F)$ and $a \in S$ for all $a \in A$ defended by S;
- $S \in gr(F)$ iff S is the least element in $co(F)$;
- $S \in pr(F)$ iff $S \in ad(F)$ and $\nexists T \in ad(F)$ s.t. $T \supset S$;
- $S \in id(F)$ iff $S \in ad(F)$, $S \subseteq \bigcap pr(F)$ and $\nexists T \in ad(F)$ s.t. $T \subseteq \bigcap pr(F)$ and $T \supset S$;
- $S \in st(F)$ iff $S \in cf(F)$ and $S^\oplus_F = A$;
- $S \in sg(F)$ iff $S \in cf(F)$ and $\nexists T \in cf(F)$ s.t. $T^\oplus_F \supset S^\oplus_F$;
- $S \in se(F)$ iff $S \in ad(F)$ and $\nexists T \in ad(F)$ s.t. $T^\oplus_F \supset S^\oplus_F$.

Given $F = (A,R)$ and a set $O \subseteq A$, the restriction of $\sigma(F)$ to O, denoted as $\sigma(F)|_O$, is the set $\{E \cap O \mid E \in \sigma(F)\}$.

Given a set of arguments A, a labelling L is a function assigning each argument $a \in A$ exactly one label among $\mathtt{t}$, $\mathtt{f}$ and $\mathtt{u}$, i.e. $L : A \mapsto \{\mathtt{t},\mathtt{f},\mathtt{u}\}$. If the arguments $A = \{a_1,\ldots,a_n\}$ are ordered, then we denote a labelling of A as a sequence of labels, e.g. the labelling $\mathtt{tuf}$ of arguments $\{a_1,a_2,a_3\}$ maps a_1 to $\mathtt{t}$, a_2 to $\mathtt{u}$, and a_3 to $\mathtt{f}$. We denote the set of all possible labellings of A as $\mathcal{L}(A)$. Likewise, given an AF F, we denote the set of all possible labellings of (the arguments of) F as $\mathcal{L}(F)$. Given a labelling L and an argument a, $L(a)$ denotes the labelling of a wrt. L; finally $in(L)$, $out(L)$, and $undec(L)$ denotes the arguments labeled to $\mathtt{t}$, $\mathtt{f}$, and $\mathtt{u}$ by L, respectively.

Definition 1. *Given a set of arguments A and labellings L_1, L_2 thereof, $L_1 \sqsubseteq L_2$ iff $in(L_1) \subseteq in(L_2)$ and $out(L_1) \subseteq out(L_2)$. Moreover we call L_1 and L_2*

- comparable *if $L_1 \sqsubseteq L_2$ or $L_2 \sqsubseteq L_1$*
- compatible *if $in(L_1) \cap out(L_2) = out(L_1) \cap in(L_2) = \emptyset$*

Note that if L_1 and L_2 are comparable then they are also compatible.

An argumentation semantics can be defined in terms of labellings rather than of extensions, i.e. the labelling-based version of a semantics σ associates to F a set $\mathcal{L}_\sigma(F) \subseteq \mathcal{L}(F)$, where any labelling $L \in \mathcal{L}_\sigma(F)$ corresponds to an extension $S \in \sigma(F)$ as follows: an argument $a \in A$ is labeled to $\mathtt{t}$ iff $a \in S$, is labeled to $\mathtt{f}$ iff $a \in S^+_F$, is labeled to $\mathtt{u}$ if neither of the above conditions holds. Given F and a set $O \subseteq A$, the restriction of $\mathcal{L}_\sigma(F)$ to O, denoted as $\mathcal{L}_\sigma(F)|_O$, is the set $\{L \cap (O \times \{\mathtt{t},\mathtt{f},\mathtt{u}\}) \mid L \in \mathcal{L}_\sigma(F)\}$. The following well-known result can be deduced e.g. from the semantics account given in [7].

Proposition 1. *For an AF F, $\mathcal{L}_{pr}(F) = \max_\sqsubseteq(\mathcal{L}_{co}(F))$.*

3 Extension-based I/O-gadgets

An I/O-gadget represents a (partial) AF where two sets of arguments are identified as input and output arguments, respectively, with the restriction that input arguments do not have any ingoing attacks.

Definition 2. *Given a set of input arguments $I \subseteq \mathfrak{A}$ and a set of output arguments $O \subseteq \mathfrak{A}$ with $I \cap O = \emptyset$, an I/O-*gadget *is an* AF $F = (A, R)$ *such that* $I, O \subseteq A$ *and* $I^-_F = \emptyset$.

The injection of a set $J \subseteq I$ to F simulates the input J in the way that all arguments in J are accepted (none of them has ingoing attacks since F is an I/O-gadget) and all arguments in $(I \setminus J)$ are rejected (each of them is attacked by the newly introduced argument z, which has no ingoing attacks).

Definition 3. *Given an I/O-gadget $F = (A, R)$ and a set of arguments $J \subseteq I$, the injection of J to F is the* AF $\triangleright(F, J) = (A \cup \{z\}, R \cup \{(z, i) \mid i \in (I \setminus J)\})$, *where z is a newly introduced argument.*

An I/O-specification describes a desired input/output behaviour by assigning to each set of input arguments a set of sets of output arguments.

Definition 4. *A* two-valued[3] I/O-specification *consists of two sets $I, O \subseteq \mathfrak{A}$ and a total function $\mathfrak{f} : 2^I \mapsto 2^{2^O}$.*

In order for an I/O-gadget F to satisfy $\mathfrak{f}$, the injection of each $J \subseteq I$ to F must have $\mathfrak{f}(J)$ as its σ-extensions restricted to the output arguments. So, informally, with input J applied the set of outputs under σ should be exactly $\mathfrak{f}(J)$.

Definition 5. *Given $I, O \subseteq \mathfrak{A}$, a semantics σ and an I/O-specification $\mathfrak{f}$, the I/O-gadget F satisfies $\mathfrak{f}$ under σ iff $\forall J \subseteq I : \sigma(\triangleright(F, J))|_O = \mathfrak{f}(J)$.*

The question we want to address is which conditions $\mathfrak{f}$ must fulfill to be satisfiable by some I/O-gadget and how an I/O-gadget doing so can be constructed.

Definition 6. *Given an I/O-specification $\mathfrak{f}$, let $Y = \{y_i \mid i \in I\}$ and $X = \{x^S_J \mid J \subseteq I, S \in \mathfrak{f}(J)\}$. The* canonical I/O-gadget *is defined as*

$$\begin{aligned}\mathfrak{C}(\mathfrak{f}) = (&I \cup O \cup Y \cup X \cup \{w\},\\ &\{(i, y_i) \mid i \in I\} \cup\\ &\{(y_i, x^S_J) \mid x^S_J \in X, i \in J\} \cup\\ &\{(i, x^S_J) \mid x^S_J \in X, i \in (I \setminus J)\} \cup\\ &\{(x_1, x_2) \mid x_1, x_2 \in X, x_1 \neq x_2\} \cup\\ &\{(x_i, w) \mid x_i \in X\} \cup \{(w, w)\} \cup\\ &\{(x^S_J, o) \mid J \subseteq I, S \in \mathfrak{f}(J), o \in (O \setminus S)\}).\end{aligned}$$

[3] In the following we omit this specification.

Intuitively, x_J^S shall enforce output S for input J. Moreover, w ensures that any stable extension of (an injection to) $\mathfrak{C}(\mathfrak{f})$ must contain an argument in X.

The following theorem shows that any I/O-specification is satisfiable under stable semantics.

Theorem 1. *Every I/O-specification $\mathfrak{f}$ is satisfied by $\mathfrak{C}(\mathfrak{f})$ under st.*

Proof. Let $I, O \subseteq \mathfrak{A}$ and $\mathfrak{f}$ be an arbitrary I/O-specification. We have to show that $st(\triangleright(\mathfrak{C}(\mathfrak{f}), J))|_O = \mathfrak{f}(J)$ holds for any $J \subseteq I$. Consider such a $J \subseteq I$.

First let $S \in \mathfrak{f}(J)$. We show that $E = \{z\} \cup J \cup \{y_i \mid i \in (I \setminus J)\} \cup \{x_J^S\} \cup S \in st(\triangleright(\mathfrak{C}(\mathfrak{f}), J))$, thus $S \in st(\triangleright(\mathfrak{C}(\mathfrak{f}), J))|_O$. E is conflict-free in $\triangleright(\mathfrak{C}(\mathfrak{f}), J)$ since z only attacks the arguments $(I \setminus J)$, an y_i with $i \in (I \setminus J)$ is only attacked by $i \notin E$, x_J^S is only attacked by other $x \in X$, $i \in (I \setminus J)$ and y_j with $j \in J$, and arguments in S are only attacked by arguments from X but not from x_J^S. E is stable in $\triangleright(\mathfrak{C}(\mathfrak{f}), J)$ since x_J^S attacks w, all other $x \in X$ and all $o \in (O \setminus S)$; z attacks all $i \in (I \setminus J)$; each y_j with $j \in J$ is attacked by j.

It remains to show that there is no $S' \in st(\triangleright(\mathfrak{C}(\mathfrak{f}), J))|_O$ with $S' \notin \mathfrak{f}(J)$. Towards a contradiction assume there is some $S' \in st(\triangleright(\mathfrak{C}(\mathfrak{f}), J))|_O$ with $S' \notin \mathfrak{f}(J)$. Hence there must be some $E' \in st(\triangleright(\mathfrak{C}(\mathfrak{f}), J))$ with $S' \subset E'$. Since w attacks itself, $w \notin E'$, thus by construction of $\mathfrak{C}(\mathfrak{f})$ there must be some $x_{J'}^{S'} \in (X \cap E')$ attacking w, and $x_{J'}^{S'}$ must attack all $o \in (O \setminus S')$. Since $S' \notin \mathfrak{f}(J)$ by assumption, it must hold that $J' \neq J$. Now note that $z \in E'$ and $j \in E'$ for all $j \in J$, since they are not attacked by construction of $\triangleright(\mathfrak{C}(\mathfrak{f}), J)$. Now if $J' \subset J$ then there is some $j \in (J \setminus J')$ attacking $x_{J'}^{S'}$, a contradiction to conflict-freeness of E'. On the other hand if $J' \not\subseteq J$ there is some $j' \in (J' \setminus J)$ which is attacked by z. Therefore also $y_{j'} \in E'$, which attacks $x_{J'}^{S'}$, again a contradiction. □

As to stage, preferred and semi-stable semantics, any I/O-specification is satisfiable, provided that a (possibly empty) output is prescribed for any input.

Proposition 2. *Every I/O-specification $\mathfrak{f}$ such that $\forall J \subseteq I, \mathfrak{f}(J) \neq \emptyset$ is satisfied by $\mathfrak{C}(\mathfrak{f})$ under $\sigma \in \{sg, pr, se\}$.*

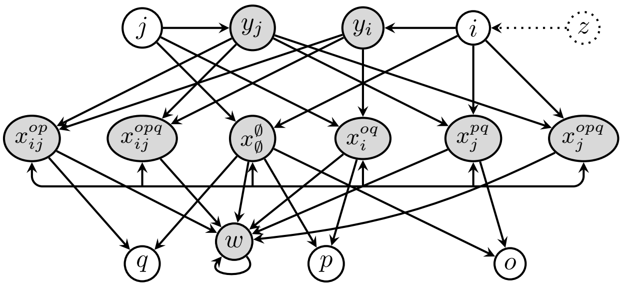
Proof (Sketch). Note that $\forall J \subseteq I$, stable, preferred, stage and semi-stable extensions coincide in $\triangleright(\mathfrak{C}(\mathfrak{f}), J)$, thus the result follows from Theorem 1. □

Theorem 2. *An I/O-specification $\mathfrak{f}$ is satisfiable under $\sigma \in \{sg, pr, se\}$ iff $\forall J \subseteq I, \mathfrak{f}(J) \neq \emptyset$.*

Proof. $\Leftarrow$: by Proposition 2.
$\Rightarrow$: Follows directly by the fact that in any AF, particularly in any injection of some extension to an I/O-gadget, a σ-extension always exists. □

Example 1. Consider the I/O-specification $\mathfrak{f}$ with $I = \{i, j\}$ and $O = \{o, p, q\}$ defined as follows: $\mathfrak{f}(\emptyset) = \{\emptyset\}$; $\mathfrak{f}(\{i\}) = \{\{o, q\}\}$; $\mathfrak{f}(\{j\}) = \{\{o, p, q\}, \{p, q\}\}$; and $\mathfrak{f}(\{i, j\}) = \{\{o, p, q\}, \{o, p\}\}$. The canonical I/O-gadget $\mathfrak{C}(\mathfrak{f})$ is depicted below[4] (without the dotted part):

[4] Argument names such as $x_{\{a,b\}}^{\{c,d\}}$ are abbreviated by x_{ab}^{cd}.



Let σ be a semantics in $\{st, sg, se, pr\}$. One can verify that for every possible input $J \subseteq I$, the injection of J to $\mathfrak{C}(\mathfrak{f})$, has exactly $\mathfrak{f}(J)$ as σ-extensions restricted to O. As an example let $J = \{j\}$. $\triangleright(\mathfrak{C}(\mathfrak{f}), \{j\})$ adds the argument z attacking i to $\mathfrak{C}(\mathfrak{f})$. Now $\sigma(\triangleright(\mathfrak{C}(\mathfrak{f}), \{j\})) = \{\{z, j, y_i, x^{\{p,q\}}_{\{j\}}, p, q\}, \{z, j, y_i, x^{\{o,p,q\}}_{\{j\}}, o, p, q\}\}$, hence $\sigma(\triangleright(\mathfrak{C}(\mathfrak{f}), \{j\}))|_O = \{\{p, q\}, \{o, p, q\}\} = \mathfrak{f}(\{j\})$.

Also for complete, grounded and ideal semantics we are able to identify a necessary and sufficient condition for satisfiability. While we show sufficiency of these conditions in more detail, their necessity is by the well-known facts that the intersection of all complete extensions is always a complete extension too and ideal and grounded semantics always yield exactly one extension.

Definition 7. *An I/O-specification $\mathfrak{f}$ is* closed *iff for each $J \subseteq I$ it holds that $\mathfrak{f}(J) \neq \emptyset$ and $\bigcap \mathfrak{f}(J) \in \mathfrak{f}(J)$.*

Proposition 3. *Every closed I/O-specification $\mathfrak{f}$ is satisfied by $\mathfrak{C}(\mathfrak{f})$ under co.*

Proof. Let $J \subseteq I$ and $\mathbb{S} = \mathfrak{f}(J)$. By construction of $\triangleright(\mathfrak{C}(\mathfrak{f}), J)$, $E^* = \{z\} \cup J \cup \{y_i \mid i \in (I \setminus J)\}$ is contained in all complete extensions, while the elements of $(I \setminus J) \cup \{y_i \mid i \in J\}$ are attacked by E^* and thus by all complete extensions. All $x^{S'}_{J'}$ with $J' \neq J$ are attacked by J or some y_i with $i \in (I \setminus J)$, thus they are attacked by E^*, while all x^S_J with $S \in \mathbb{S}$ attack each other and the other attacks they receive come from elements attacked by E^* in turn. Two cases can then be distinguished. If $|\mathbb{S}| = 1$ then by construction of $\triangleright(\mathfrak{C}(\mathfrak{f}), J)$ there is just one x^S_J defended by E^*, thus the only complete extension is $E^* \cup \{x^S_J\} \cup S$. If, on the other hand, $|\mathbb{S}| > 1$, any x^S_J with $S \in \mathbb{S}$ can be included, giving rise to the complete extension $E^* \cup \{x^S_J\} \cup S$, or none of x^S_J can be included, giving rise to the complete extension $E^* \cup \bigcap \mathbb{S}$ since an x^S_J attacks all $o \in (O \setminus S)$. Taking into account that $\bigcap \mathbb{S} \in \mathbb{S}$, in both cases we have that $co(\triangleright(F, J))|_O = \mathfrak{f}(J)$. □

Theorem 3. *An I/O-specification $\mathfrak{f}$ is satisfiable under co iff $\mathfrak{f}$ is closed.*

Proposition 4. *Every I/O-specification $\mathfrak{f}$ with $|\mathfrak{f}(J)| = 1$ for each $J \subseteq I$ is satisfied by $\mathfrak{C}(\mathfrak{f})$ under gr and id.*

Proof (Sketch). This follows the same idea as the proof of Proposition 3. Since $|\mathfrak{f}(J)| = 1$, $\triangleright(\mathfrak{C}(\mathfrak{f}), J)$ has only one complete extension for each $J \subseteq I$, which is also grounded and ideal. □

Theorem 4. *An I/O-specification $\mathfrak{f}$ is satisfiable under gr and id iff $|\mathfrak{f}(J)| = 1$ for each $J \subseteq I$.*

4 Labelling-based I/O-gadgets

In the previous section we have dealt with I/O-specifications mapping extensions to sets of extensions. In general, there are two reasons why an argument does not belong to an extension, i.e. either because it is attacked by the extension or because it is undecided due to insufficient justification. This distinction impacts on the justification status of arguments, since attacks from undecided arguments can prevent attacked arguments to belong to an extension, while attacks from out arguments are ineffective. In order to take into account this distinction, we first provide a labelling-based counterpart of the notions introduced in Section 3.

Definition 8. *A* 3-valued I/O-specification *consists of two sets $I, O \subseteq \mathfrak{A}$ and a total function $\mathfrak{f} : \mathcal{L}(I) \mapsto 2^{\mathcal{L}(O)}$.*

Definition 9. *Given an I/O-gadget $F = (A, R)$ and a labelling L of I, the labelling-injection of L to F is the* AF $\blacktriangleright(F, L) = (A \cup \{z\}, R \cup \{(z, a) \mid L(a) = f\} \cup \{(b, b) \mid L(b) = \mathtt{u}\})$*, where z is a newly introduced argument.*

Definition 10. *Given $I, O \subseteq \mathfrak{A}$, a semantics σ and a 3-valued I/O-specification $\mathfrak{f}$, the I/O-gadget F satisfies $\mathfrak{f}$ under σ iff $\forall L \in \mathcal{L}(I) : \mathcal{L}_\sigma(\blacktriangleright(F, L))|_O = \mathfrak{f}(L)$.*

By definition of stable labellings it is clear that in order to be satisfied under *st*, a 3-valued I/O-specification must only be non-empty for labellings with each argument labeled to $\mathtt{t}$ or $\mathtt{f}$.

Theorem 5. *A 3-valued I/O-specification $\mathfrak{f}$ is satisfiable under st iff for each $L \in \mathcal{L}(I)$ it holds that*

- *if $\exists i \in I : L(i) = \mathtt{u}$ then $\mathfrak{f}(L) = \emptyset$, and*
- *otherwise $K(o) \neq \mathtt{u}$ for all $K \in \mathfrak{f}(L)$ and $o \in O$.*

Proof. $\Rightarrow$: If $\exists i \in I : L(i) = \mathtt{u}$ then for any I/O-gadget F, $\blacktriangleright(F, L)$ contains a self-attacking argument otherwise unattacked, hence $\mathcal{L}_{st}(\blacktriangleright(F, L)) = \emptyset$. In the other case, by definition of stable extension it is clear that each $o \in O$ must be labelled either $\mathtt{t}$ or $\mathtt{f}$ by any stable labelling.
$\Leftarrow$: Follows from Theorem 1. If $\exists i \in I : L(i) = \mathtt{u}$ then $st(\blacktriangleright(\mathfrak{C}(\mathfrak{f}), L)) = \emptyset$, otherwise the labelling-injection coincides with the normal injection. □

In order to characterize those 3-valued I/O-specifications which are satisfiable under the other semantics we need the concept of monotonicity.

Definition 11. *A 3-valued I/O-specification $\mathfrak{f}$ is* monotonic *iff for all L_1 and L_2 such that $L_1 \sqsubseteq L_2$ it holds that $\forall K_1 \in \mathfrak{f}(L_1) \exists K_2 \in \mathfrak{f}(L_2) : K_1 \sqsubseteq K_2$.*

The intuitive meaning of monotonicity is the following: if K_1 is an output for input L_1, then for every input which is more informative than L_1 there must be an output more informative than K_1. First a rather obvious observation:

Proposition 5. *For every 3-valued I/O-specification $\mathfrak{f}$ which is satisfiable under gr, $|\mathfrak{f}(L)| = 1$ for all $L \in \mathcal{L}(I)$.*

The following was shown in Proposition 7 of [1] for complete semantics.

Proposition 6. *Every 3-valued I/O-specification which is satisfiable under {gr, pr} is monotonic.*

Proof (Sketch). Let $\mathfrak{f}$ be a 3-valued I/O-specification and suppose it is satisfied by the I/O-gadget F under *gr*. Moreover let $L_1 \sqsubseteq L_2$ be labellings of I. Proposition 7 of [1] says that $\forall K_2 \in \mathcal{L}_{co}(\blacktriangleright(F, L_2))|_O \, \exists K_1 \in \mathcal{L}_{co}(\blacktriangleright(F, L_1))|_O : K_1 \sqsubseteq K_2$ and $\forall K_1 \in \mathcal{L}_{co}(\blacktriangleright(F, L_1))|_O \, \exists K_2 \in \mathcal{L}_{co}(\blacktriangleright(F, L_2))|_O : K_1 \sqsubseteq K_2$. By the facts that $|\mathfrak{f}(L_1)|{=}|\mathfrak{f}(L_2)|{=}1$ (cf. Proposition 5) and $\mathcal{L}_{pr}(F) = \max_{\sqsubseteq}(\mathcal{L}_{co}(F))$ for each AF F (cf. Proposition 1), the result follows for *gr* and *pr*, respectively. □

In Propositions 5 and 6 we have given necessary conditions for 3-valued I/O-specifications to be satisfiable under *gr* and *pr*. In the following we show that these conditions are also sufficient in the sense that we can find a satisfying I/O-gadget. The constructions of these I/O-gadgets will depend on the given 3-valued I/O-specification and on the semantics, but they will share the same input and output part. The semantics-specific part, denoted by $X^\sigma_\mathfrak{f}$ and $R^\sigma_\mathfrak{f}$ in the following definition, will be given later.

Definition 12. *Given a 3-valued I/O-specification* $\mathfrak{f}$ *we define* $I' = \{i' \mid i \in I\}$, $O' = \{o' \mid o \in O\}$, $R_I = \{(i, i') \mid i \in I\}$ *and* $R_O = \{(o', o'), (o', o) \mid o \in O\}$. *The* 3-valued canonical I/O-gadget *for semantics* σ *is defined as*

$$\mathfrak{D}^\sigma_\mathfrak{f} = (I \cup I' \cup X^\sigma_\mathfrak{f} \cup O' \cup O, R_I \cup R^\sigma_\mathfrak{f} \cup R_O).$$

with $R^\sigma_\mathfrak{f} \subseteq ((I \cup I') \times X^\sigma_\mathfrak{f}) \cup (X^\sigma_\mathfrak{f} \times X^\sigma_\mathfrak{f}) \cup (X^\sigma_\mathfrak{f} \times (O' \cup O))$.

Now we turn to the semantics-specific constructions. For grounded semantics we need the concept of *determining* input labellings. An input labelling L is determining for output argument o if L is a minimal (w.r.t. $\sqsubseteq$) input labelling where o gets a concrete value ($\mathtt{t}$ or $\mathtt{f}$) according to $\mathfrak{f}$.

With abuse of notation, in the following we may identify a set including a single labelling with the labelling itself.

Definition 13. *Given a 3-valued I/O-specification* $\mathfrak{f}$ *with* $|\mathfrak{f}(L)| = 1$ *for all* $L \in \mathcal{L}(I)$ *and an argument* $o \in O$, *a labelling* L *of* I *is determining for* o, *if* $\mathfrak{f}(L)(o) \neq \mathtt{u}$ *and* $\forall L' \sqsubsetneq L : \mathfrak{f}(L')(o) = \mathtt{u}$. *We denote the set of labellings which are determining for* o *as* $\mathfrak{d}_\mathfrak{f}(o)$.

Example 2. Let $\mathfrak{f}$ be the following 3-valued I/O-specification with $I = \{i_1, i_2\}$ and $O = \{o_1, o_2\}$: $\mathfrak{f}(\mathtt{uu}) = \{\mathtt{uu}\}$; $\mathfrak{f}(\mathtt{tu}) = \{\mathtt{tu}\}$; $\mathfrak{f}(\mathtt{ut}) = \{\mathtt{ut}\}$; $\mathfrak{f}(\mathtt{uf}) = \{\mathtt{uf}\}$; $\mathfrak{f}(\mathtt{fu}) = \{\mathtt{uu}\}$; $\mathfrak{f}(\mathtt{tt}) = \{\mathtt{tt}\}$; $\mathfrak{f}(\mathtt{tf}) = \{\mathtt{tf}\}$; $\mathfrak{f}(\mathtt{ft}) = \{\mathtt{ut}\}$; and $\mathfrak{f}(\mathtt{ff}) = \{\mathtt{tf}\}$ We have the following sets of determining labellings: $\mathfrak{d}_\mathfrak{f}(o_1) = \{\mathtt{tu}, \mathtt{ff}\}$ and $\mathfrak{d}_\mathfrak{f}(o_2) = \{\mathtt{ut}, \mathtt{uf}\}$. Consider, for instance, the input labelling $\mathtt{ff}$. We have $\mathfrak{f}(\mathtt{ff}) = \mathtt{tf}$. In order to check if $\mathtt{ff}$ is determining for o_1 we have to look at all input labellings being less committed than $\mathtt{ff}$. Now we observe $\mathfrak{f}(\mathtt{uf}) = \mathtt{uf}$, $\mathfrak{f}(\mathtt{fu}) = \mathfrak{f}(\mathtt{uu}) = \mathtt{uu}$. In all of these desired output labellings o_1 has value $\mathtt{u}$, so $\mathtt{ff}$ is determining for o_1. On the other hand $\mathtt{ff}$ is not determining for o_2, since $\mathfrak{f}(\mathtt{uf})(o_2) = \mathtt{f}$.

Definition 14. *Given a* 3*-valued* I/O*-specification* $\mathfrak{f}$ *with* $|\mathfrak{f}(L){=}1|$ *for all* $L \in \mathcal{L}(I)$*, the gr-specific part of* $\mathfrak{D}_{\mathfrak{f}}^{gr}$ *is given by*

$$X_{\mathfrak{f}}^{gr} = \{x_o^L \mid o \in O, L \in \mathfrak{d}_{\mathfrak{f}}(o)\}, \text{ and}$$

$$\begin{aligned} R_{\mathfrak{f}}^{gr} = &\{(i, x_o^L) \mid x_o^L \in X_{\mathfrak{f}}^{gr}, L(i) = \mathtt{f}\} \cup \{(i', x_o^L) \mid x_o^L \in X_{\mathfrak{f}}^{gr}, L(i) = \mathtt{t}\} \cup \\ &\{(x_o^L, o') \mid x_o^L \in X_{\mathfrak{f}}^{gr}, \mathfrak{f}(L)(o) = \mathtt{t}\} \cup \{(x_o^L, o) \mid x_o^L \in X_{\mathfrak{f}}^{gr}, \mathfrak{f}(L)(o) = \mathtt{f}\}. \end{aligned}$$

For every $o \in O$ and each labelling L which is determining for o, there is the argument x_o^L. This argument can be labelled $\mathtt{t}$ if L is the labelling of I and intuitively enforces the labelling of o to be as given by $\mathfrak{f}(L)$.

The next results, requiring two preliminary lemmata, characterize satisfiability of grounded semantics.

Lemma 1. *Let* $\mathfrak{f}$ *be a* 3*-valued* I/O*-specification which is monotonic and s.t.* $|\mathfrak{f}(L)| = 1$ *for each* $L \in \mathcal{L}(I)$*. Let* $o \in O$ *and* $L, L' \in \mathcal{L}(I)$ *such that* $\mathfrak{f}(L)(o) = \mathtt{t}$ *and* $\mathfrak{f}(L')(o) = \mathtt{f}$*. Then* L *and* L' *are not compatible.*

Lemma 2. *Given a* 3*-valued* I/O*-specification* $\mathfrak{f}$ *which is monotonic and s.t.* $|\mathfrak{f}(L)| = 1$ *for each* $L \in \mathcal{L}(I)$*, let* $o \in O$ *and* $L, L' \in \mathcal{L}(I)$ *such that* L *is determining for* o*. Then* $\mathcal{L}_{gr}(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{gr}, L'))(x_o^L)$ *is*

- $\mathtt{t}$ *iff* $L \sqsubseteq L'$*;*
- $\mathtt{f}$ *iff* L *and* L' *are not compatible; and*
- $\mathtt{u}$ *iff* $L \not\sqsubseteq L'$ *but* L *and* L' *are compatible.*

Proposition 7. *Every* 3*-valued* I/O*-specification* $\mathfrak{f}$ *which is monotonic and s.t.* $|\mathfrak{f}(L)| = 1$ *for each* $L \in \mathcal{L}(I)$*, is satisfied by* $\mathfrak{D}_{\mathfrak{f}}^{gr}$ *under gr.*

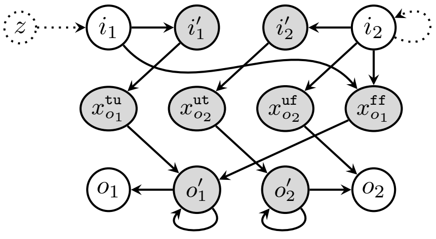
Proof. Consider some input labelling L. We have to show $\mathcal{L}_{gr}(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{gr}, L))|_O = \mathfrak{f}(L)$. To this end let $o \in O$.

Assume $\mathfrak{f}(L)(o) = \mathtt{u}$. Then, since $\mathfrak{f}$ is monotonic, $\mathfrak{f}(L')(o) = \mathtt{u}$ for all $L' \sqsubseteq L$. Therefore, there is no $L' \sqsubseteq L$ with $L' \in \mathfrak{d}_{\mathfrak{f}}(o)$. By Lemma 2 we get that for all $L'' \in \mathfrak{d}_{\mathfrak{f}}(o)$ it holds that $\mathcal{L}_{gr}(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{gr}, L))(x_o^{L''}) \neq \mathtt{t}$. Since such $x_o^{L''}$ are the only potential attackers of o and o', $\mathcal{L}_{gr}(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{gr}, L))(o) = \mathtt{u}$.

Next assume $\mathfrak{f}(L)(o) = \mathtt{t}$. Then there is some $L' \sqsubseteq L$ with $L' \in \mathfrak{d}_{\mathfrak{f}}(o)$ and $\mathfrak{f}(L')(o) = \mathtt{t}$. By Lemma 2 we get $\mathcal{L}_{gr}(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{gr}, L))(x_o^{L'}) = \mathtt{t}$. Moreover, $x_o^{L'}$ attacks o', hence $\mathcal{L}_{gr}(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{gr}, K))(o') = \mathtt{f}$. Towards a contradiction assume there is some $x_o^{L''}$ attacking o with $\mathcal{L}_{gr}(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{gr}, L))(x_o^{L''}) \in \{\mathtt{t}, \mathtt{u}\}$. Then, by Lemma 2, L'' and L are compatible. However, by construction of $\mathfrak{D}_{\mathfrak{f}}^{gr}$, $\mathfrak{f}(L'')(o) = \mathtt{f}$, $\mathfrak{f}(L)(o) = \mathtt{t}$ and, by Lemma 1 L'' and L are not compatible.

Finally assume $\mathfrak{f}(L)(o) = \mathtt{f}$. Then there is some $L' \sqsubseteq L$ with $L' \in \mathfrak{d}_{\mathfrak{f}}(o)$ and $\mathfrak{f}(L')(o) = \mathtt{f}$. By Lemma 2 we get $\mathcal{L}_{gr}(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{gr}, L))(x_o^{L'}) = \mathtt{t}$. Moreover, $x_o^{L'}$ attacks o, hence $\mathcal{L}_{gr}(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{gr}, L))(o) = \mathtt{f}$. □

Example 3. Again consider the 3-valued I/O-specification $\mathfrak{f}$ from Example 2. We have seen the determining labellings there. The I/O-gadget $\mathfrak{D}_{\mathfrak{f}}^{gr}$ is depicted below. Consider for example the labelling-injection of $\mathtt{fu}$ to $\mathfrak{D}_{\mathfrak{f}}^{gr}$, which is indicated by the dotted part of the figure. We get $\mathcal{L}_{gr}(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{gr}, \mathtt{fu}))|_O = \mathtt{uu}$, satisfying the I/O-specification. One can check that this holds for all possible labelling-injections, hence $\mathfrak{D}_{\mathfrak{f}}^{gr}$ satisfies $\mathfrak{f}$ under the grounded semantics.



Theorem 6. *A* 3*-valued I/O-specification* $\mathfrak{f}$ *is satisfiable under gr iff* $\mathfrak{f}$ *is monotonic and for each* $L \in \mathcal{L}(I)$*,* $|\mathfrak{f}(L)| = 1$*.*

Definition 15. *Given a* 3*-valued I/O-specification* $\mathfrak{f}$*, the pr-specific part of* $\mathfrak{D}_{\mathfrak{f}}^{pr}$ *is given by*

$$
\begin{aligned}
X_{\mathfrak{f}}^{pr} =& \{x_K^L \mid L \in \mathcal{L}(I), K \in \mathfrak{f}(L)\}, \text{ and}\\
R_{\mathfrak{f}}^{pr} =& \{(i, x_K^L) \mid x_K^L \in X_{\mathfrak{f}}^{pr}, L(i) = \mathit{f}\} \cup \{(i', x_K^L) \mid x_K^L \in X_{\mathfrak{f}}^{pr}, L(i) = \mathit{t}\} \cup\\
& \{(x_K^L, o') \mid x_K^L \in X_{\mathfrak{f}}^{pr}, K(o) = \mathit{t}\} \cup \{(x_K^L, o) \mid x_K^L \in X_{\mathfrak{f}}^{pr}, K(o) = \mathit{f}\} \cup\\
& \{(x_K^L, x_{K'}^{L'}) \mid \neg(L \sqsubset L' \land K \sqsubseteq K') \land \neg(L \sqsupset L' \land K \sqsupseteq K')\}.
\end{aligned}
$$

Every input-output-combination is represented by an argument in $\mathfrak{D}_{\mathfrak{f}}^{pr}$. We first show a technical lemma, giving sufficient conditions on the labelling-status of the arguments in $X_{\mathfrak{f}}^{pr}$ to get the desired labelling of the output arguments.

Lemma 3. *Given a* 3*-valued I/O-specification* $\mathfrak{f}$ *and an input labelling* $L \in \mathcal{L}(I)$*. The following holds for each preferred labelling P of* $\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{pr}, L)$*: If* $P(x_K^L) = \mathtt{t}$ *and for all* $x_{K'}^{L'} \in X_{\mathfrak{f}}^{pr}$*,* $K' \sqsupset K \Rightarrow P(x_{K'}^{L'}) \neq \mathtt{t}$ *and* $(K' \not\sqsubseteq K \land K \not\sqsubseteq K') \Rightarrow P(x_{K'}^{L'}) = \mathit{f}$*, then* $P|_O = K$*.*

We proceed by showing that $\mathfrak{D}^{pr}$ satisfies every monotonic function under the preferred semantics.

Proposition 8. *Every* 3*-valued I/O-specification* $\mathfrak{f}$ *which is monotonic is satisfied by* $\mathfrak{D}_{\mathfrak{f}}^{pr}$ *under pr.*

Proof. Consider an arbitrary input labelling $L \in \mathcal{L}(I)$. We have to show that $\mathcal{L}_{pr}(\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{pr}, L))|_O = \mathfrak{f}(L)$.

Similar to Lemma 2 one can check that those $x_{K'}^{L'} \in X_{\mathfrak{f}}^{pr}$ with $L \sqsupseteq L'$ are the only arguments in $X_{\mathfrak{f}}^{pr}$ which can be $\mathtt{t}$ in a preferred labelling of $\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{pr}, L)$. Now the arguments x_K^L with $K \in \mathfrak{f}(L)$ form a clique in $\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{pr}, L)$. Moreover each of these x_K^L defends itself, hence there is a preferred labelling of $\blacktriangleright(\mathfrak{D}_{\mathfrak{f}}^{pr}, L)$ for each $K \in \mathfrak{f}(L)$ identified by x_K^L. Let P_K be the preferred labelling with

$P_K(x^L_K) = \mathtt{t}$ where $K \in \mathfrak{f}(L)$. All $x^{L'}_{K'}$ with $K' \not\sqsubseteq K \wedge K' \not\sqsupseteq K$ are attacked by x^K_L, hence $P_K(x^{L'}_{K'}) = \mathtt{f}$. Assume $K' \sqsupset K$. If $L' \not\sqsupset L$, then $x^{L'}_{K'}$ is again attacked by x^K_L and $P_K(x^{L'}_{K'}) = \mathtt{f}$. If $L' \sqsupset L$, $P_K(x^{L'}_{K'}) \neq \mathtt{t}$ since it is attacked by some i or i' $(i \in I)$ which is $\mathtt{u}$ in P_K. Therefore, by Lemma 3, $P_K|_O = K$.

It remains to show that there is no other preferred labelling besides these P_K with $K \in \mathfrak{f}(L)$. Towards a contradiction, assume that there is a preferred labelling P' where no x^L_K with $K \in \mathfrak{f}(L)$ is $\mathtt{t}$. By our initial considerations, those $x^{L'}_{K'}$ with $L' \sqsubset L$ are the only x-arguments which can be $\mathtt{t}$ in P'. It cannot be the case that none of them is $\mathtt{t}$, since P' would not be preferred. Then there is at least an $x^{L'}_{K'}$ which is $\mathtt{t}$ in P', with $L' \sqsubset L$, and without loss of generality we can assume that there is no $x^{L''}_{K''}$ which is $\mathtt{t}$ and $L' \sqsubset L''$. Now, since $\mathfrak{f}$ is monotonic there has to be a $K \in \mathfrak{f}(L)$ such that $K' \sqsubseteq K$. We prove that no x-argument attacking x^L_K is $\mathtt{t}$ in P'.

First, the only x-arguments that can be $\mathtt{t}$ are those $x^{L''}_{K''}$ with $L'' \sqsubset L$. Note that, according to Definition 15, $x^{L'}_{K'}$ does not attack x^L_K, since $L' \sqsubset L \wedge K' \sqsubseteq K$. If an attacker $x^{L''}_{K''}$ is attacked in turn by $x^{L'}_{K'}$ then it is $\mathtt{f}$, otherwise either $L'' \sqsubset L' \wedge K'' \sqsubseteq K'$ or $L' \sqsubset L'' \wedge K' \sqsubseteq K''$. The first case is impossible, since we would have $L'' \sqsubset L \wedge K'' \sqsubseteq K$, entailing that $x^{L''}_{K''}$ does not attack x^L_K. In the other case, by the assumption on $x^{L'}_{K'}$ it holds that $x^{L''}_{K''}$ is not $\mathtt{t}$.

Now, x^L_K defends itself against all x-arguments and none of them is $\mathtt{t}$, moreover by construction of $\blacktriangleright(\mathfrak{D}^{pr}_{\mathfrak{f}}, L)$, all attackers from I and I' are $\mathtt{f}$. But then, consider the labelling P'' obtained from P' by assigning to x^L_K the label $\mathtt{t}$, and by assigning to all the attackers of x^L_K the label $\mathtt{f}$. P'' is admissible and $P' \sqsubseteq P''$, contradicting the maximality of P'. □

Theorem 7. *A* 3*-valued* I/O*-specification* $\mathfrak{f}$ *is satisfiable under pr iff* $\mathfrak{f}$ *is monotonic.*

As to remaining semantics, note that Propositions 7 and 8 apply to ideal and semi-stable semantics, respectively, since for each L the grounded labelling of $\blacktriangleright(\mathfrak{D}^{gr}_{\mathfrak{f}}, L)$ coincides with the ideal labelling, and the preferred labellings of $\blacktriangleright(\mathfrak{D}^{pr}_{\mathfrak{f}}, L)$ coincide with semi-stable labellings. However, this does not allow to derive a complete characterization, since there are non-monotonic 3-valued I/O-specifications, satisfiable by ideal and semi-stable semantics, respectively.

5 Conclusions

To the best of our knowledge, this is the first characterization of the input/output expressive power of argumentation semantics. In [10], expressiveness has been studied as the capability of enforcing sets of extensions. The problem faced in this paper differs in two aspects: on the one hand, we have to enforce a set of extensions for any input rather than a single set of extensions, on the other hand we can exploit non-output arguments that are not seen outside a subframework. Moreover we also consider labellings besides extensions. A labelling-based investigation exploiting hidden arguments is carried out in [11], but still in the context of an ordinary AF rather than an I/O-gadget.

We restricted our considerations to total I/O-specifications, where the output is defined for each input. One can also think of situations where we do not care about the output for some inputs, i.e. the interest lies in the satisfiability of a partial function. We did not tackle these issues here, but plan to do so as part of future work.

Further future work includes the 3-valued I/O-characterization of complete semantics, being the only transparent semantics [1] for which this was left open. Moreover, the investigation of further semantics such as CF2 [4] would be of interest. Another issue is the construction of I/O-gadgets from *compact* I/O-specifications where the function is not explicitly stated but, for instance, described as a Boolean (or three-valued) circuit. We conjecture that I/O-gadgets can then be composed from simple building blocks along the lines of the given circuit. A related question in this direction is the identification of minimal I/O-gadgets satisfying a given specification.

References

1. Baroni, P., Boella, G., Cerutti, F., Giacomin, M., van der Torre, L., Villata, S.: On the Input/Output behaviour of argumentation frameworks. Artif. Intell. 217, 144–197 (2014)
2. Baroni, P., Caminada, M., Giacomin, M.: An introduction to argumentation semantics. Knowledge Engineering Review 26(4), 365–410 (2011)
3. Baroni, P., Giacomin, M.: On principle-based evaluation of extension-based argumentation semantics. Artif. Intell. 171(10-15), 675–700 (2007)
4. Baroni, P., Giacomin, M., Guida, G.: SCC-Recursiveness: A general schema for argumentation semantics. Artif. Intell. 168(1-2), 162–210 (2005)
5. Caminada, M., Amgoud, L.: On the evaluation of argumentation formalisms. Artif. Intell. 171(5-6), 286–310 (2007)
6. Caminada, M., Carnielli, W.A., Dunne, P.E.: Semi-stable semantics. Journal of Logic and Computation 22(5), 1207–1254 (2012)
7. Caminada, M., Gabbay, D.M.: A logical account of formal argumentation. Studia Logica 93(2), 109–145 (2009)
8. Diller, M., Haret, A., Linsbichler, T., Rümmele, S., Woltran, S.: An extension-based approach to belief revision in abstract argumentation. In: Proc. IJCAI. pp. 2926–2932. AAAI Press (2015)
9. 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), 321–357 (1995)
10. Dunne, P.E., Dvořák, W., Linsbichler, T., Woltran, S.: Characteristics of multiple viewpoints in abstract argumentation. In: Proc. KR. pp. 72–81. AAAI Press (2014)
11. Dyrkolbotn, S.K.: How to argue for anything: Enforcing arbitrary sets of labellings using AFs. In: Proc. KR. pp. 626–629. AAAI Press (2014)
12. Modgil, S., Bench-Capon, T.J.M.: Metalevel argumentation. Journal of Logic and Computation 21(6), 959–1003 (2011)
13. Verheij, B.: Two approaches to dialectical argumentation: admissible sets and argumentation stages. In: Proc. NAIC. pp. 357–368 (1996)