On the Functional Completeness of Argumentation Semantics

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

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

April 29, 2016

Introduction

- m . . . abstract argumentation is the most important topic in AI
- a . . . abstract argumentation has no real-world applications
- b . . . machine learning is more successful than KR

Introduction

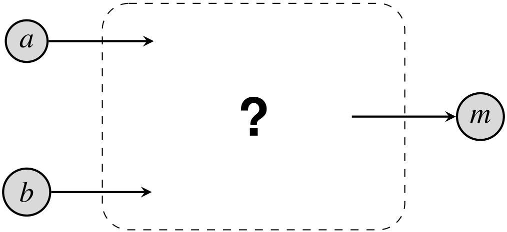
- m . . . abstract argumentation is the most important topic in AI
- a . . . abstract argumentation has no real-world applications
- b . . . machine learning is more successful than KR
- We want to argue for m as long as not both a, b turn out to be true.

Introduction

- m . . . abstract argumentation is the most important topic in AI
- a . . . abstract argumentation has no real-world applications
- b . . . machine learning is more successful than KR

- We want to argue for m as long as not both a, b turn out to be true.

$\Rightarrow$ Is there an argumentation framework such that, under a certain semantics σ, m is accepted if at most one of a and b is accepted and m is not accepted otherwise?



Introduction

- Abstract Argumentation Frameworks (AFs) with designated input- and output-arguments.
- Question: Which functions from input assignments to (multiple) output assignments are realizable by such AFs?
- Exact characterization of realizable functions.

Introduction

- Abstract Argumentation Frameworks (AFs) with designated input- and output-arguments.
- Question: Which functions from input assignments to (multiple) output assignments are realizable by such AFs?
- Exact characterization of realizable functions.

- Connecting two recent lines of research in abstract argumentation.
  - Realizability [Dunne et al., 2015]: Capabilities of semantics in terms of expressiveness.
  - Input/Output-AFs [Baroni et al., 2014]: Decomposability and transparency of semantics.

Introduction

- Abstract Argumentation Frameworks (AFs) with designated input- and output-arguments.
- Question: Which functions from input assignments to (multiple) output assignments are realizable by such AFs?
- Exact characterization of realizable functions.
- Connecting two recent lines of research in abstract argumentation.
  - Realizability [Dunne et al., 2015]: Capabilities of semantics in terms of expressiveness.
  - Input/Output-AFs [Baroni et al., 2014]: Decomposability and transparency of semantics.
- Adds to the systematic comparison of semantics [Baroni and Giacomin, 2007].
- Strategic argumentation: Deciding whether achieving a certain goal is possible and, if yes, how to do so.

Outline

- Background
- Realizability
- Input/Output-AFs

- I/O-characterization of extension-based semantics
- I/O-characterization of labelling-based semantics

- Conclusion

Background

Definition

An argumentation framework (AF) is a pair (A, R) where

- $A \subseteq \mathfrak{A}$ is a finite set of arguments and
- $R \subseteq A \times A$ is the attack relation representing conflicts.

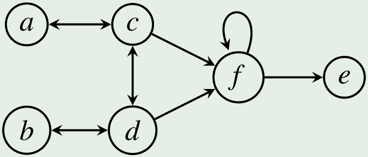
Background

Definition

An argumentation framework (AF) is a pair (A, R) where

- $A \subseteq \mathfrak{A}$ is a finite set of arguments and
- $R \subseteq A \times A$ is the attack relation representing conflicts.

Example



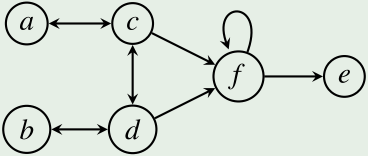
$$F = \big(\{a, b, c, d, e, f\},$$
$$\{(a, c), (c, a), (c, d), (d, c), (d, b), (b, d), (c, f), (d, f), (f, f), (f, e)\}\big)$$

Background (ctd.)

Conflict-free Sets

Given an AF $F = (A, R)$, a set $S \subseteq A$ is conflict-free in F, if, for each $a, b \in S$, $(a, b) \notin R$.

Example



$$\mathit{cf}(F) = \{\{a,b,e\},\{a,d,e\},\{b,c,e\},\\ \{a,b\},\{a,d\},\{a,e\},\{b,c\},\{b,e\},\{d,e\},\{c,e\},\\ \{a\},\{b\},\{c\},\{d\},\{e\},\emptyset\}$$

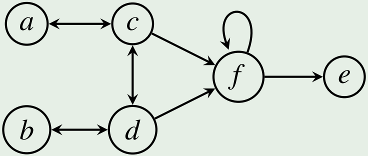
Background (ctd.)

Admissible Sets

Given an AF $F = (A, R)$, a set $S \subseteq A$ is admissible in F, if

- S is conflict-free in F and
- each $a \in S$ is defended by S in F, i.e. for each $b \in A$ with $(b, a) \in R$, there exists some $c \in S$, such that $(c, b) \in R$.

Example



adm$(F) = \{$ ~~$\{a, b, e\}$~~, $\{a, d, e\}, \{b, c, e\},$
$\{a, b\}, \{a, d\},$ ~~$\{a, e\}$~~, $\{b, c\},$ ~~$\{b, e\}$~~, $\{d, e\}, \{c, e\},$
$\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \emptyset \}$

Background (ctd.)

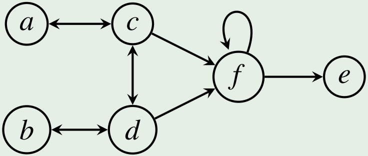
Preferred Semantics

Given an AF $F = (A, R)$, a set $S \subseteq A$ is a preferred extension in F, if

- S is admissible in F and
- there is no admissible $T \subseteq A$ with $T \supset S$.

$\Rightarrow$ Maximal admissible sets (w.r.t. set-inclusion).

Example



prf$(F) = \{$ ~~$\{a, b, e\}$~~, $\{a, d, e\}, \{b, c, e\}$,
$\{a, b\}$, ~~$\{a, d\}, \{a, e\}, \{b, c\}, \{b, e\}, \{d, e\}, \{c, e\},$~~
~~$\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \emptyset$~~$\}$

Background (ctd.)

Further semantics:

- Complete semantics
- Grounded semantics
- Stage semantics [Verheij, 1996]
- Semi-stable semantic [Caminada et al., 2012]
- Ideal semantics [Dung et al., 2007]

- cf2 semantics [Baroni et al., 2005, Gaggl and Woltran, 2013]
- Resolution-based grounded semantics [Baroni et al., 2011]
- ...

Background (ctd.)

Labelling-based semantics

- More fine-grained evaluation of AFs. [Caminada and Gabbay, 2009]
- A labelling is a function assigning each argument one label among $\mathtt{t}$, $\mathtt{f}$, and $\mathtt{u}$.

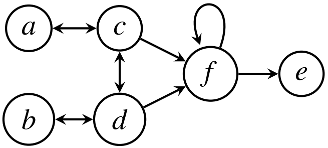
Definition

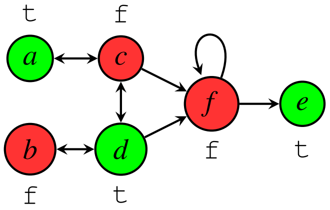
The labelling-based version of a semantics σ associates to an AF $F = (A, R)$ a set $\mathcal{L}_\sigma(F)$, where any labelling $L \in \mathcal{L}_\sigma(F)$ corresponds to an extension $E \in \sigma(F)$ as follows:

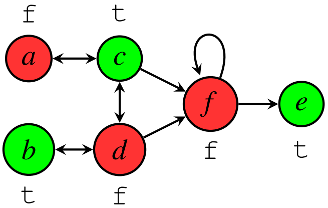
- $L(a) = \mathtt{t}$ iff $a \in E$;
- $L(a) = \mathtt{f}$ iff $\exists b \in E : (b, a) \in R$;
- $L(a) = \mathtt{u}$ iff neither of the above holds.

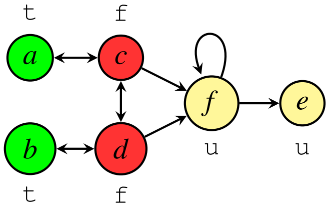
Background (ctd.)

Labelling-based semantics









Realizability [Dunne et al., 2015]

Definition

Given a semantics σ, a set $\mathbb{S} \subseteq 2^{\mathfrak{A}}$ is realizable under σ if there exists an AF having $\sigma(F) = \mathbb{S}$.

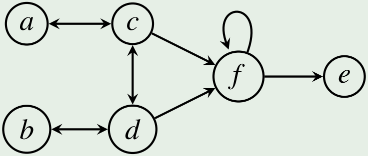
Realizability [Dunne et al., 2015]

Definition

Given a semantics σ, a set $\mathbb{S} \subseteq 2^{\mathfrak{A}}$ is realizable under σ if there exists an AF having $\sigma(F) = \mathbb{S}$.

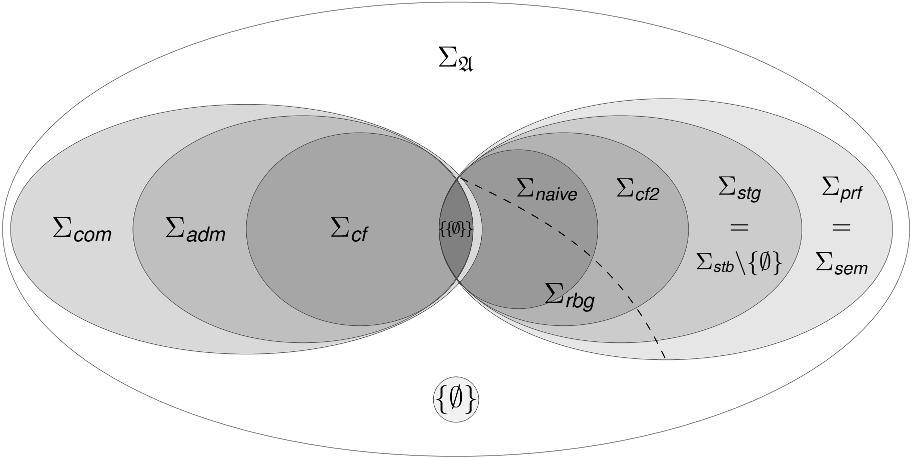
Example

$\mathbb{S} = \{\{a, b\}, \{a, d, e\}, \{b, c, e\}\}$.



- $\mathbb{S}$ is realizable under *prf*, since *prf*$(F) = \mathbb{S}$.
- $\mathbb{S}$ is *not* realizable under *stb*.

Realizability [Dunne et al., 2015]

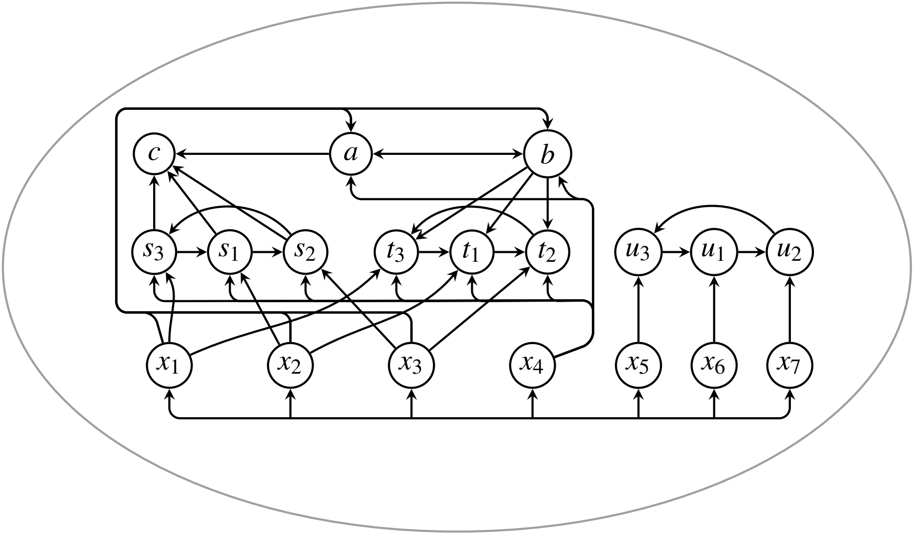


$\Sigma_\sigma = \{\sigma(F) \mid F = (A, R)$ is an AF$\}$.

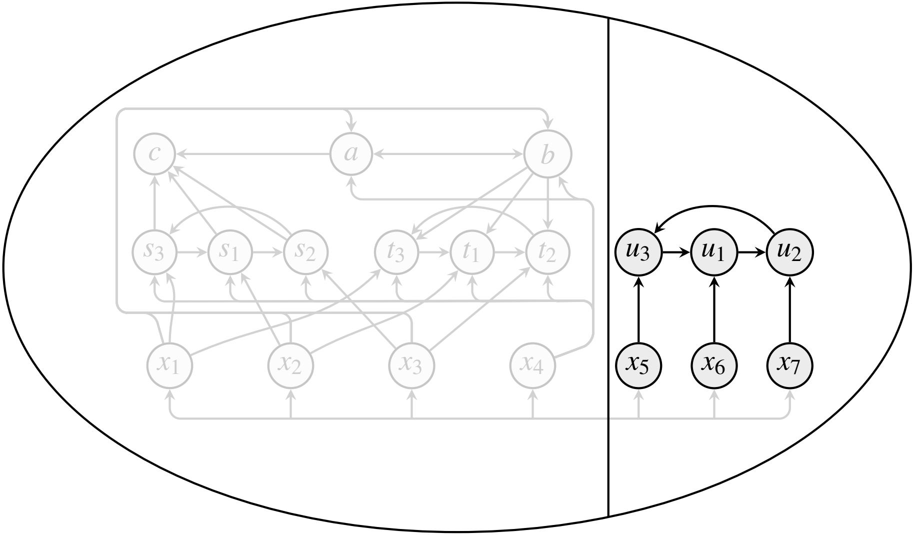
Input/Output AFs [Baroni et al., 2014]

- Investigation of the input/output-behaviour of argumentation semantics
- Decomposability: Given an arbitrary partition of an AF, can the extensions under σ be determined by composition of the partial evaluations?
  - Allows for incremental computation.
- Transparency: Can parts of AFs be replaced by components which are input/output-equivalent under σ?
  - Allows for summarization, i.e. hiding irrelevant parts of big AFs.

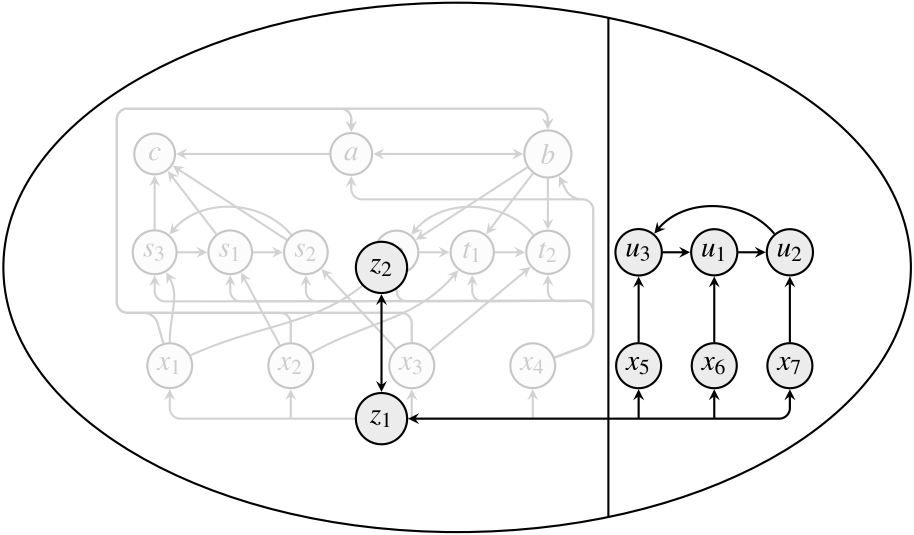
Input/Output AFs [Baroni et al., 2014]



Input/Output AFs [Baroni et al., 2014]



Input/Output AFs [Baroni et al., 2014]



Input/Output AFs [Baroni et al., 2014]

- Investigation of the input/output-behaviour of argumentation semantics
- Decomposability: Given an arbitrary partition of an AF, can the extensions under σ be determined by composition of the partial evaluations?
  - Allows for incremental computation.
- Transparency: Can parts of AFs be replaced by components which are input/output-equivalent under σ?
  - Allows for summarization, i.e. hiding irrelevant parts of big AFs.

	stb	*prf*	*com*	*grd*	*sem*	*id*
Decomposability	Yes	No	Yes	No	No	No
SCC-Decomposability	Yes	Yes	Yes	Yes	No	No
Transparency	Yes	No*	Yes	Yes	No	No
SCC-Transparency	Yes	Yes	Yes	Yes	No	No

* Yes under additional mild conditions

Extension-based I/O-characterization

Definition

An I/O-specification consists of two sets $I, O \subseteq \mathfrak{A}$ and a total function $\mathfrak{p} : 2^I \mapsto 2^{2^O}$.

Example

Input ($I = \{a, b\}$)	Output ($O = \{m\}$)
$\emptyset$	$\{\{m\}\}$
$\{a\}$	$\{\{m\}\}$
$\{b\}$	$\{\{m\}\}$
$\{a, b\}$	$\{\emptyset\}$

Extension-based I/O-characterization

Definition

An I/O-specification consists of two sets $I, O \subseteq \mathfrak{A}$ and a total function $\mathfrak{p} : 2^I \mapsto 2^{2^O}$.

Example

Input ($I = \{a, b\}$)	Output ($O = \{m\}$)
$\emptyset$	$\{\{m\}\}$
$\{a\}$	$\{\{m\}\}$
$\{b\}$	$\{\{m\}\}$
$\{a, b\}$	$\{\emptyset\}$

Question

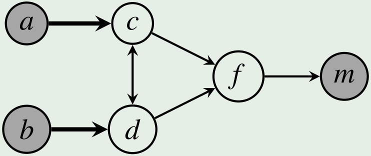
Given an I/O-specification $\mathfrak{p} : 2^I \mapsto 2^{2^O}$, is $\mathfrak{p}$ satisfiable?

Extension-based I/O-characterization

Definition

Given input arguments I and output arguments O with $I \cap O = \emptyset$, an I/O-gadget is an AF $F = (A, R)$ such that $I, O \subseteq A$ and $I_F^- = \emptyset$.

Example



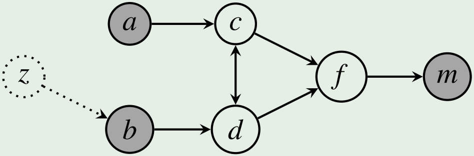
Extension-based I/O-characterization

Definition

Given an I/O-gadget $F = (A, R)$ the injection of $J \subseteq I$ to F is the AF $\triangleright(F, J) = (A \cup \{z\}, R \cup \{(z, i) \mid i \in (I \setminus J)\})$.

Example

Injection of $\{a\}$:



Extension-based I/O-characterization

Definition

An I/O-specification consists of two sets $I, O \subseteq \mathfrak{A}$ and a total function $\mathfrak{p} : 2^I \mapsto 2^{2^O}$.

Example

Input ($I = \{a, b\}$)	Output ($O = \{m\}$)
$\emptyset$	$\{\{m\}\}$
$\{a\}$	$\{\{m\}\}$
$\{b\}$	$\{\{m\}\}$
$\{a, b\}$	$\{\emptyset\}$

Extension-based I/O-characterization

Definition

An I/O-specification consists of two sets $I, O \subseteq \mathfrak{A}$ and a total function $\mathfrak{p} : 2^I \mapsto 2^{2^O}$.

Example

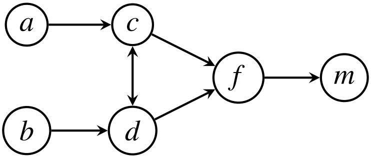
Input ($I = \{a, b\}$)	Output ($O = \{m\}$)
$\emptyset$	$\{\{m\}\}$
$\{a\}$	$\{\{m\}\}$
$\{b\}$	$\{\{m\}\}$
$\{a, b\}$	$\{\emptyset\}$

Definition

The I/O-gadget F satisfies I/O-specification $\mathfrak{p}$ under semantics σ iff $\forall J \subseteq I : \sigma(\triangleright(F, J))|_O = \mathfrak{p}(J)$.

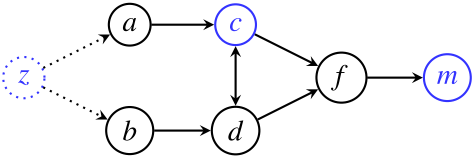
Extension-based I/O-characterization

Input ($I = \{a, b\}$)	Output ($O = \{m\}$)
$\emptyset$	$\{\{m\}\}$
$\{a\}$	$\{\{m\}\}$
$\{b\}$	$\{\{m\}\}$
$\{a, b\}$	$\{\emptyset\}$



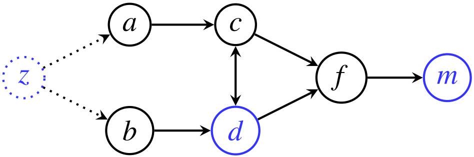
Extension-based I/O-characterization

Input ($I = \{a, b\}$)	Output ($O = \{m\}$)
$\emptyset$	$\{\{m\}\}$
$\{a\}$	$\{\{m\}\}$
$\{b\}$	$\{\{m\}\}$
$\{a, b\}$	$\{\emptyset\}$



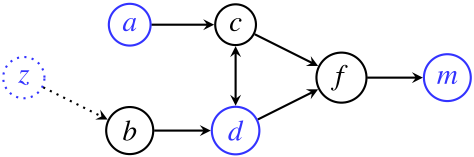
Extension-based I/O-characterization

Input ($I = \{a, b\}$)	Output ($O = \{m\}$)
$\emptyset$	$\{\{m\}\}$
$\{a\}$	$\{\{m\}\}$
$\{b\}$	$\{\{m\}\}$
$\{a, b\}$	$\{\emptyset\}$



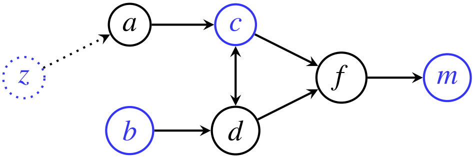
Extension-based I/O-characterization

Input ($I = \{a, b\}$)	Output ($O = \{m\}$)
$\emptyset$	$\{\{m\}\}$
$\{a\}$	$\{\{m\}\}$
$\{b\}$	$\{\{m\}\}$
$\{a, b\}$	$\{\emptyset\}$



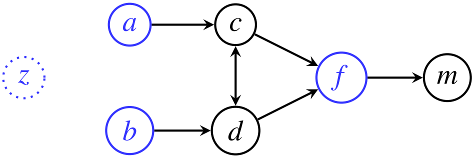
Extension-based I/O-characterization

Input ($I = \{a, b\}$)	Output ($O = \{m\}$)
$\emptyset$	$\{\{m\}\}$
$\{a\}$	$\{\{m\}\}$
$\{b\}$	$\{\{m\}\}$
$\{a, b\}$	$\{\emptyset\}$



Extension-based I/O-characterization

Input ($I = \{a, b\}$)	Output ($O = \{m\}$)
$\emptyset$	$\{\{m\}\}$
$\{a\}$	$\{\{m\}\}$
$\{b\}$	$\{\{m\}\}$
$\{a, b\}$	$\{\emptyset\}$



Extension-based I/O-characterization

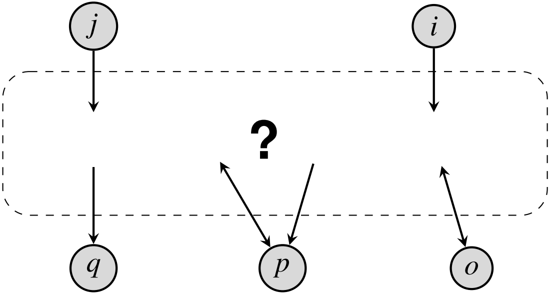
Another I/O-specification:

Input ($I = \{i,j\}$)	Output ($O = \{o,p,q\}$)
$\emptyset$	$\{\emptyset\}$
$\{i\}$	$\{\{o,q\}\}$
$\{j\}$	$\{\{o,p,q\},\{p,q\}\}$
$\{i,j\}$	$\{\{o,p,q\},\{o,p\}\}$

Extension-based I/O-characterization

Another I/O-specification:

Input ($I = \{i, j\}$)	Output ($O = \{o, p, q\}$)
$\emptyset$	$\{\emptyset\}$
$\{i\}$	$\{\{o, q\}\}$
$\{j\}$	$\{\{o, p, q\}, \{p, q\}\}$
$\{i, j\}$	$\{\{o, p, q\}, \{o, p\}\}$



Extension-based I/O-characterization

Definition

Given I/O-specification $\mathfrak{p}$, let $X = \{x_J^S \mid J \subseteq I, S \in \mathfrak{p}(J)\}$ and $Y = \{y_i \mid i \in I\}$. The canonical I/O-gadget (for $\mathfrak{p}$) is defined as

$$\begin{aligned}\mathcal{C}(\mathfrak{p}) = (&I \cup O \cup Y \cup X \cup \{w\},\\ &\{(i, y_i) \mid i \in I\} \cup\\ &\{(y_i, x_J^S) \mid x_J^S \in X, i \in J\} \cup\\ &\{(i, x_J^S) \mid x_J^S \in X, i \in (I \setminus J)\} \cup\\ &\{(x, x') \mid x, x' \in X, x \neq x'\} \cup\\ &\{(x, w) \mid x \in X\} \cup \{(w, w)\} \cup\\ &\{(x_J^S, o) \mid x_J^S \in X, o \in (O \setminus S)\}).\end{aligned}$$

Extension-based I/O-characterization

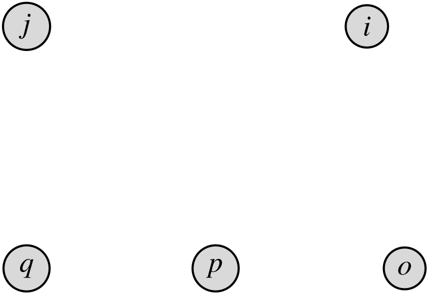
Theorem

An I/O-specification $\mathfrak{p}$ is satisfiable under σ iff

stb:	$\top$
prf, *sem*, *stg*:	$\forall J \subseteq I : \|\mathfrak{p}(J)\| \geq 1$
com:	$\forall J \subseteq I : \|\mathfrak{p}(J)\| \geq 1 \wedge \bigcap \mathfrak{p}(J) \in \mathfrak{p}(J)$
grd, *id*:	$\forall J \subseteq I : \|\mathfrak{p}(J)\| = 1$

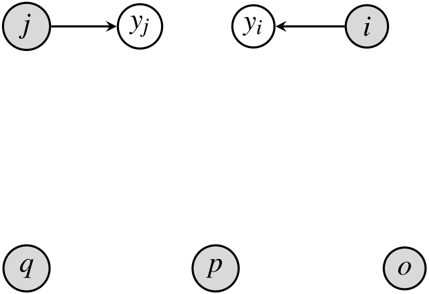
Extension-based I/O-characterization

Input ($I = \{i, j\}$)	Output ($O = \{o, p, q\}$)
$\emptyset$	$\{\emptyset\}$
$\{i\}$	$\{\{o, q\}\}$
$\{j\}$	$\{\{o, p, q\}, \{p, q\}\}$
$\{i, j\}$	$\{\{o, p, q\}, \{o, p\}\}$



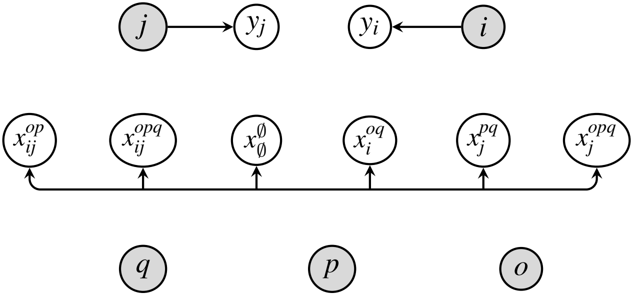
Extension-based I/O-characterization

Input ($I = \{i,j\}$)	Output ($O = \{o,p,q\}$)
$\emptyset$	$\{\emptyset\}$
$\{i\}$	$\{\{o,q\}\}$
$\{j\}$	$\{\{o,p,q\},\{p,q\}\}$
$\{i,j\}$	$\{\{o,p,q\},\{o,p\}\}$



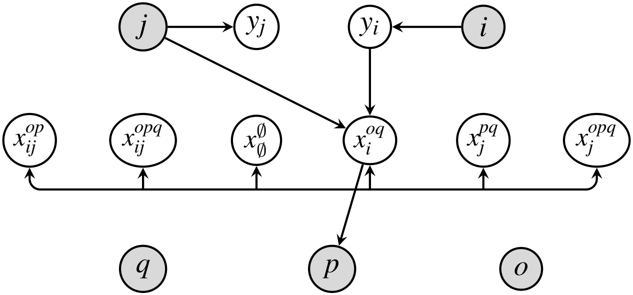
Extension-based I/O-characterization

Input ($I = \{i, j\}$)	Output ($O = \{o, p, q\}$)
$\emptyset$	$\{\emptyset\}$
$\{i\}$	$\{\{o, q\}\}$
$\{j\}$	$\{\{o, p, q\}, \{p, q\}\}$
$\{i, j\}$	$\{\{o, p, q\}, \{o, p\}\}$



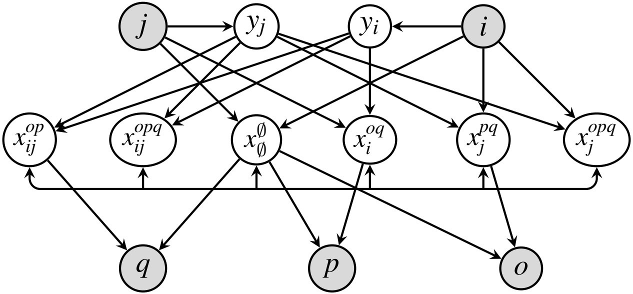
Extension-based I/O-characterization

Input ($I = \{i, j\}$)	Output ($O = \{o, p, q\}$)
$\emptyset$	$\{\emptyset\}$
$\{i\}$	$\{\{o, q\}\}$
$\{j\}$	$\{\{o, p, q\}, \{p, q\}\}$
$\{i, j\}$	$\{\{o, p, q\}, \{o, p\}\}$



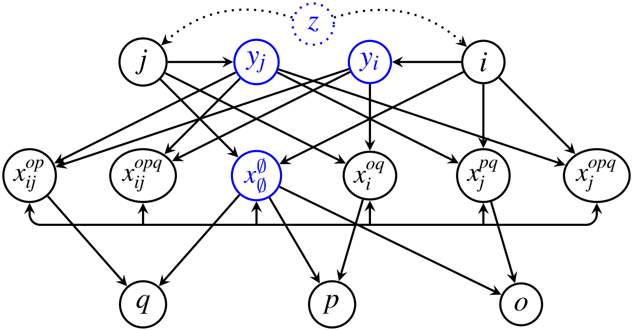
Extension-based I/O-characterization

Input ($I = \{i, j\}$)	Output ($O = \{o, p, q\}$)
$\emptyset$	$\{\emptyset\}$
$\{i\}$	$\{\{o, q\}\}$
$\{j\}$	$\{\{o, p, q\}, \{p, q\}\}$
$\{i, j\}$	$\{\{o, p, q\}, \{o, p\}\}$



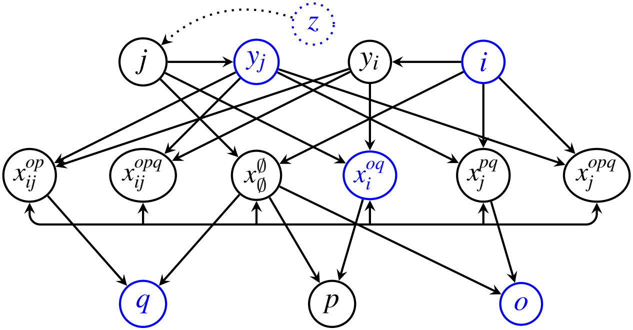
Extension-based I/O-characterization

Input ($I = \{i,j\}$)	Output ($O = \{o,p,q\}$)
$\emptyset$	$\{\emptyset\}$
$\{i\}$	$\{\{o,q\}\}$
$\{j\}$	$\{\{o,p,q\},\{p,q\}\}$
$\{i,j\}$	$\{\{o,p,q\},\{o,p\}\}$



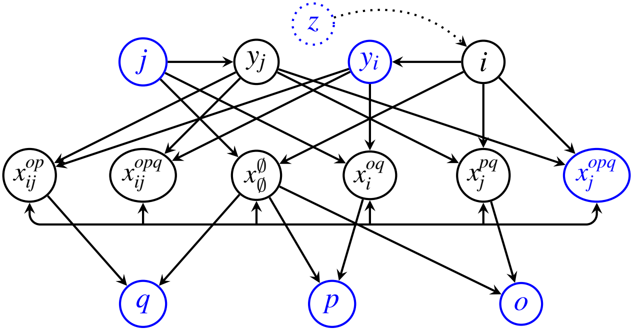
Extension-based I/O-characterization

Input ($I = \{i, j\}$)	Output ($O = \{o, p, q\}$)
$\emptyset$	$\{\emptyset\}$
$\{i\}$	$\{\{o, q\}\}$
$\{j\}$	$\{\{o, p, q\}, \{p, q\}\}$
$\{i, j\}$	$\{\{o, p, q\}, \{o, p\}\}$



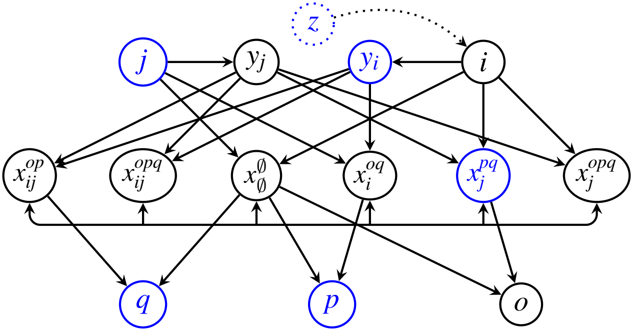
Extension-based I/O-characterization

Input ($I = \{i, j\}$)	Output ($O = \{o, p, q\}$)
$\emptyset$	$\{\emptyset\}$
$\{i\}$	$\{\{o, q\}\}$
$\{j\}$	$\{\{o, p, q\}, \{p, q\}\}$
$\{i, j\}$	$\{\{o, p, q\}, \{o, p\}\}$



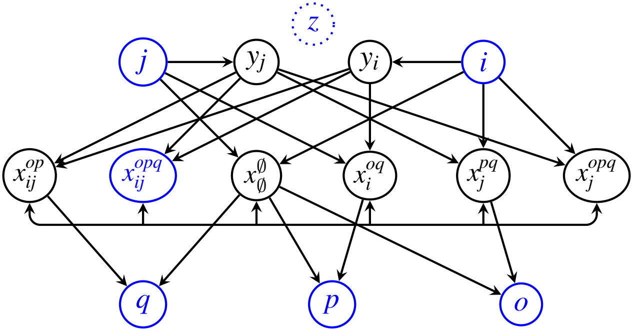
Extension-based I/O-characterization

Input ($I = \{i, j\}$)	Output ($O = \{o, p, q\}$)
$\emptyset$	$\{\emptyset\}$
$\{i\}$	$\{\{o, q\}\}$
$\{j\}$	$\{\{o, p, q\}, \{p, q\}\}$
$\{i, j\}$	$\{\{o, p, q\}, \{o, p\}\}$



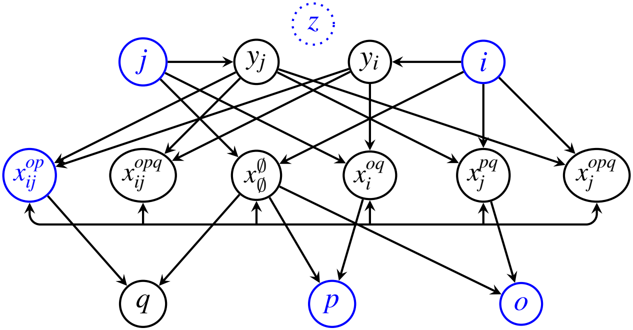
Extension-based I/O-characterization

Input ($I = \{i, j\}$)	Output ($O = \{o, p, q\}$)
$\emptyset$	$\{\emptyset\}$
$\{i\}$	$\{\{o, q\}\}$
$\{j\}$	$\{\{o, p, q\}, \{p, q\}\}$
$\{i, j\}$	$\{\{o, p, q\}, \{o, p\}\}$



Extension-based I/O-characterization

Input ($I = \{i, j\}$)	Output ($O = \{o, p, q\}$)
$\emptyset$	$\{\emptyset\}$
$\{i\}$	$\{\{o, q\}\}$
$\{j\}$	$\{\{o, p, q\}, \{p, q\}\}$
$\{i, j\}$	$\{\{o, p, q\}, \{o, p\}\}$



Labelling-based I/O-characterization

Definition

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

Input ($I = \{i,j\}$)	Output ($O = \{o,p\}$)
$\{i \leftarrow \mathtt{u}, j \leftarrow \mathtt{u}\}$	$\{\{o \leftarrow \mathtt{u}, p \leftarrow \mathtt{u}\}\}$
$\{i \leftarrow \mathtt{t}, j \leftarrow \mathtt{u}\}$	$\{\{o \leftarrow \mathtt{t}, p \leftarrow \mathtt{u}\}\}$
$\{i \leftarrow \mathtt{u}, j \leftarrow \mathtt{t}\}$	$\{\{o \leftarrow \mathtt{u}, p \leftarrow \mathtt{t}\}\}$
$\{i \leftarrow \mathtt{f}, j \leftarrow \mathtt{u}\}$	$\{\{o \leftarrow \mathtt{u}, p \leftarrow \mathtt{u}\}\}$
$\{i \leftarrow \mathtt{u}, j \leftarrow \mathtt{f}\}$	$\{\{o \leftarrow \mathtt{u}, p \leftarrow \mathtt{f}\}\}$
$\{i \leftarrow \mathtt{t}, j \leftarrow \mathtt{t}\}$	$\{\{o \leftarrow \mathtt{t}, p \leftarrow \mathtt{f}\}\}$
$\{i \leftarrow \mathtt{t}, j \leftarrow \mathtt{f}\}$	$\{\{o \leftarrow \mathtt{t}, p \leftarrow \mathtt{f}\}\}$
$\{i \leftarrow \mathtt{f}, j \leftarrow \mathtt{t}\}$	$\{\{o \leftarrow \mathtt{u}, p \leftarrow \mathtt{t}\}\}$
$\{i \leftarrow \mathtt{f}, j \leftarrow \mathtt{f}\}$	$\{\{o \leftarrow \mathtt{t}, p \leftarrow \mathtt{f}\}\}$

Definition

The I/O-gadget F satisfies the 3-valued I/O-specification $\mathfrak{p}$ under semantics σ iff $\forall L \subseteq \mathcal{L}(I) : \mathcal{L}_\sigma(\blacktriangleright(F,L))|_O = \mathfrak{p}(L)$.

Labelling-based I/O-characterization

Definition

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

Example

Input ($I = \{i, j\}$)	Output ($O = \{o, p\}$)
$\{i \leftarrow \mathtt{u}, j \leftarrow \mathtt{u}\}$	$\{\{o \leftarrow \mathtt{u}, p \leftarrow \mathtt{u}\}\}$
$\{i \leftarrow \mathtt{t}, j \leftarrow \mathtt{u}\}$	$\{\{o \leftarrow \mathtt{u}, p \leftarrow \mathtt{u}\}, \{o \leftarrow \mathtt{u}, p \leftarrow \mathtt{f}\}\}$
$\vdots$	$\vdots$
$\{i \leftarrow \mathtt{t}, j \leftarrow \mathtt{t}\}$	$\{\{o \leftarrow \mathtt{u}, p \leftarrow \mathtt{u}\}\}$
$\{i \leftarrow \mathtt{t}, j \leftarrow \mathtt{f}\}$	$\{\{o \leftarrow \mathtt{t}, p \leftarrow \mathtt{f}\}, \{o \leftarrow \mathtt{t}, p \leftarrow \mathtt{t}\}\}$
$\vdots$	$\vdots$

Labelling-based I/O-characterization

Theorem

A 3-valued I/O-specification $\mathfrak{p}$ is satisfiable under σ iff

- *stb*: $\forall L \in \mathcal{L}(I) \forall K \in \mathfrak{p}(L) \forall o \in O : K(o) \neq \mathtt{u}$
- *prf*: $\mathfrak{p}$ is monotonic
- *grd*: $\mathfrak{p}$ is monotonic and $\forall L \subseteq \mathcal{L}(I) : |\mathfrak{p}(L)| = 1$

Conclusion

Summary

- First step toward a combination of recent lines of research.
  - Input/Output argumentation frameworks.
  - Realizability of argumentation semantics.
- I/O-characterizations: Exact conditions for satisfiability.
  - Extension-based: most prominent semantics.
  - Labelling-based: preferred, stable and grounded semantics.
- Constructions for satisfiable I/O-specifications.
- Characterizations for partial I/O-specifications.

Summary

- First step toward a combination of recent lines of research.
  - Input/Output argumentation frameworks.
  - Realizability of argumentation semantics.
- I/O-characterizations: Exact conditions for satisfiability.
  - Extension-based: most prominent semantics.
  - Labelling-based: preferred, stable and grounded semantics.
- Constructions for satisfiable I/O-specifications.
- Characterizations for partial I/O-specifications.

Future Work

- 3-valued I/O-characterization of complete semantics.
- Construction of I/O-gadgets from compact representations of I/O-specifications, such as Boolean (resp. 3-valued) formulas or circuits.
- Identification of minimal I/O-gadgets.

References I

Baroni, P., Boella, G., Cerutti, F., Giacomin, M., van der Torre, L., and Villata, S. (2014).
On the Input/Output behaviour of argumentation frameworks.
Artif. Intell., 217:144–197.

Baroni, P., Dunne, P. E., and Giacomin, M. (2011).
On the resolution-based family of abstract argumentation semantics and its grounded instance.
Artif. Intell., 175(3-4):791–813.

Baroni, P. and Giacomin, M. (2007).
On principle-based evaluation of extension-based argumentation semantics.
Artif. Intell., 171(10-15):675–700.

Baroni, P., Giacomin, M., and Guida, G. (2005).
SCC-Recursiveness: A general schema for argumentation semantics.
Artif. Intell., 168(1-2):162–210.

References II

Baumann, R., Dvořák, W., Linsbichler, T., Strass, H., and Woltran., S. (2014).
Compact argumentation frameworks.
In Proc. ECAI, pages 69–74.

Bench-Capon, T. J. M. and Dunne, P. E. (2007).
Argumentation in artificial intelligence.
Artif. Intell., 171(10-15):619–641.

Caminada, M., Carnielli, W. A., and Dunne, P. E. (2012).
Semi-stable semantics.
Journal of Logic and Computation, 22(5):1207–1254.

Caminada, M. and Gabbay, D. M. (2009).
A logical account of formal argumentation.
Studia Logica, 93(2):109–145.

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

References III

- Dung, P. M., Mancarella, P., and Toni, F. (2007).
  Computing ideal sceptical argumentation.
  Artif. Intell., 171(10-15):642–674.
- Dunne, P. E., Dvořák, W., Linsbichler, T., and Woltran, S. (2015).
  Characteristics of multiple viewpoints in abstract argumentation.
  Artif. Intell., 228:153–178.
- Dyrkolbotn, S. K. (2014).
  How to argue for anything: Enforcing arbitrary sets of labellings using AFs.
  In Proc. KR, pages 626–629. AAAI Press.
- Gaggl, S. A. and Woltran, S. (2013).
  The cf2 argumentation semantics revisited.
  Journal of Logic and Computation, 23(5):925–949.
- Verheij, B. (1996).
  Two approaches to dialectical argumentation: admissible sets and argumentation stages.
  In Proc. NAIC, pages 357–368.