FAKULTÄT FÜR INFORMATIK
Faculty of Informatics

On the Functional Completeness of Argumentation Semantics

Massimiliano Giacomin, Thomas Linsbichler, Stefan Woltran

5th Workshop on Dynamics of Knowledge and Belief and 4th Workshop KI & Kognition

Dresden, September 22, 2015

Introduction

- Argumentation has become a major topic in AI research.
- Gives answers to “how assertions are proposed, discussed, and resolved in the context of issues upon which several diverging opinions may be held” [Bench-Capon and Dunne, 2007].
- Connections to other AI formalism: knowledge representation, nonmonotonic reasoning, multiagent systems.
- Dung’s Abstract Argumentation Frameworks (AFs) [Dung, 1995] conceal the concrete contents of arguments; only consider the conflict between them $\Rightarrow$ attack graph.
- Argumentation semantics: rules for identifying sets of acceptable arguments.
- Recent years have seen some work on structural analysis of their capabilities. [Dunne et al., 2015, Dyrkolbotn, 2014]

Functional Completeness

- 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.

Functional Completeness

- 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.

Functional Completeness

- 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 comparision of semanitcs [Baroni and Giacomin, 2007].
- Modular and dynamic aspects of argumentation.
- 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

Countably infinite set of arguments $\mathfrak{A}$.

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

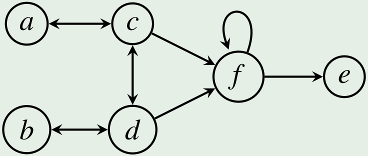
Countably infinite set of arguments $\mathfrak{A}$.

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

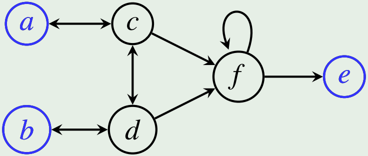
Given an AF $F = (A, R)$, a set $S \subseteq A$ is conflict-free in F, if, for each $a, b \in S$, $(a, b) \notin R$.

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

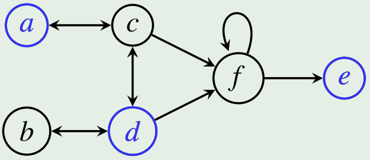


cf$(F) = \{\{a, b, e\},$

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



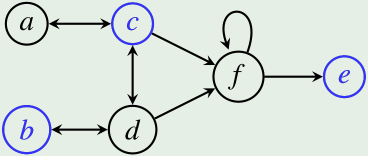
cf$(F) = \{\{a, b, e\}, \{a, d, e\},$

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



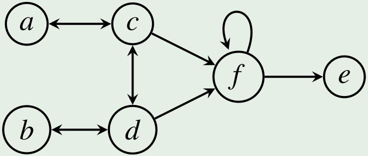
$cf(F) = \{\{a, b, e\}, \{a, d, e\}, \{b, c, e\},$

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



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

Background (ctd.)

Stable Extensions

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

- S is conflict-free in F and
- for each $a \in A \setminus S$, there exists some $b \in S$, such that $(b, a) \in R$.

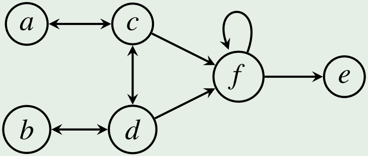
Background (ctd.)

Stable Extensions

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

- S is conflict-free in F and
- for each $a \in A \setminus S$, there exists some $b \in S$, such that $(b, a) \in R$.

Example



stb$(F) = \{\sout{\{a, b, e\}},$

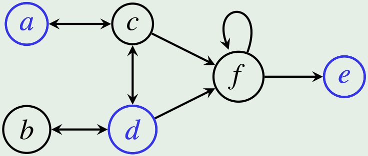
Background (ctd.)

Stable Extensions

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

- S is conflict-free in F and
- for each $a \in A \setminus S$, there exists some $b \in S$, such that $(b, a) \in R$.

Example



stb$(F) = \{\{\cancel{a, b, e}\}, \{a, d, e\},$

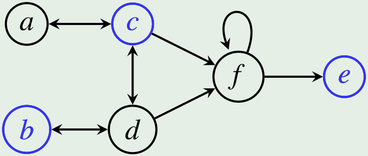
Background (ctd.)

Stable Extensions

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

- S is conflict-free in F and
- for each $a \in A \setminus S$, there exists some $b \in S$, such that $(b, a) \in R$.

Example



stb$(F) = \{\sout{\{a, b, e\}}, \{a, d, e\}, \{b, c, e\}\}$

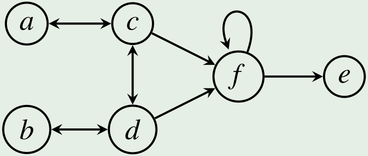
Background (ctd.)

Stable Extensions

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

- S is conflict-free in F and
- for each $a \in A \setminus S$, there exists some $b \in S$, such that $(b, a) \in R$.

Example



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

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 $(c, b) \in R$.

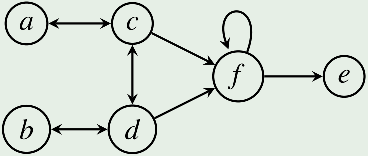
Background (ctd.)

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$(F) = \{$~~$\{a, b, e\}$~~,

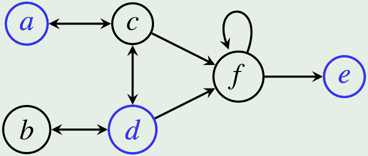
Background (ctd.)

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

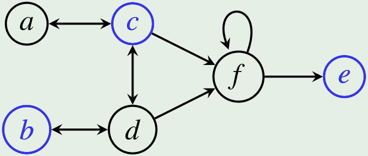
Background (ctd.)

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



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

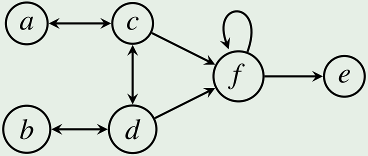
Background (ctd.)

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

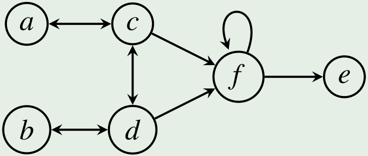
Background (ctd.)

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

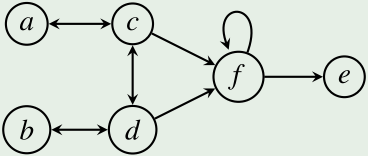
Background (ctd.)

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 Extensions

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).

Background (ctd.)

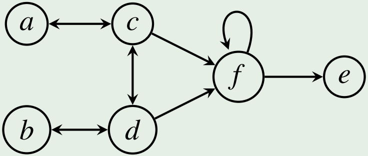
Preferred Extensions

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:

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

- 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 $\mathtt{u}$.

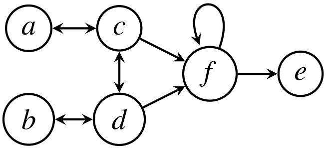
Definition

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:

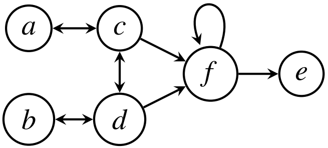
- $L(a) = \mathtt{t}$ iff $a \in E$;
- $L(a) = \mathtt{f}$ iff $\exists b \in E : (b, a) \in R$;
- $L(a) = \mathtt{u}$ iff neither of the above holds.

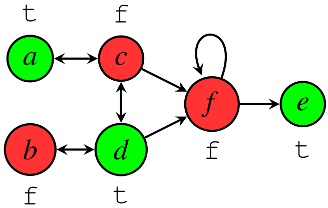
Background (ctd.)

Labelling-based semantics

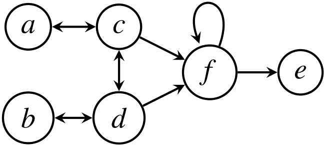


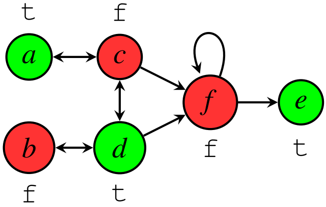
Labelling-based semantics

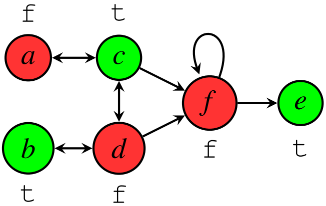




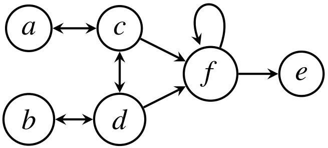
Labelling-based semantics

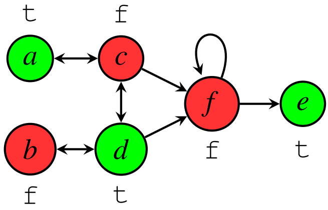


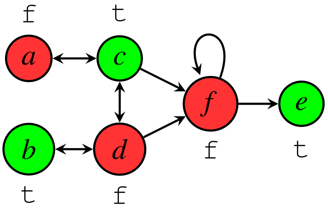


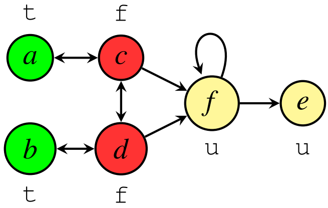


Labelling-based semantics









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 $\sigma(F) = \mathbb{S}$.

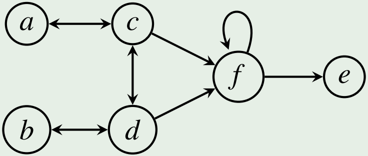
Realizability [Dunne et al., 2015]

Definition

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

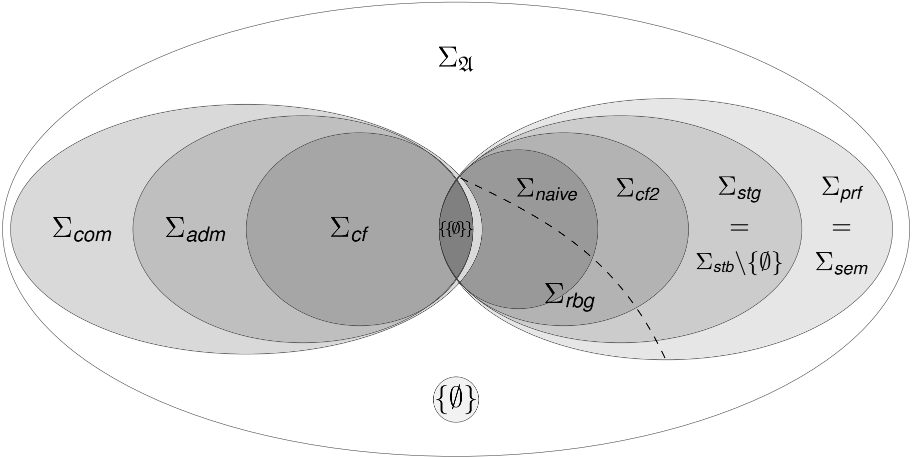
Example

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



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

Realizability [Dunne et al., 2015]

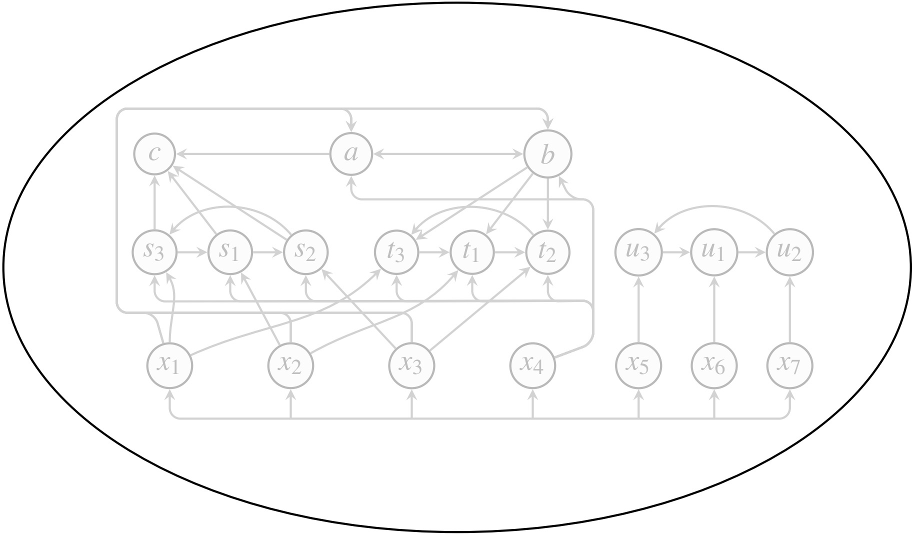


$\Sigma_\sigma = \{\sigma(F) \mid F = (A, R)$ is an AF$\}$.

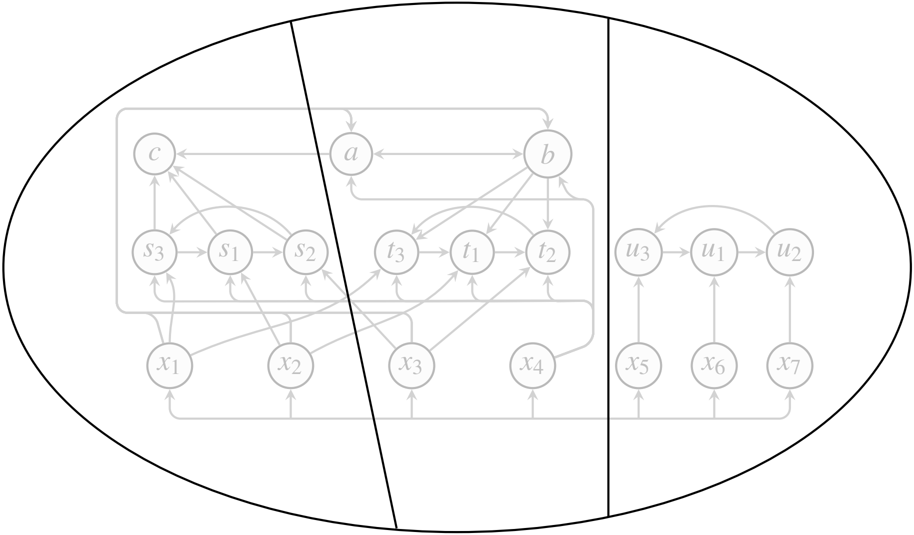
Input/Output AFs [Baroni et al., 2014]

- 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.

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



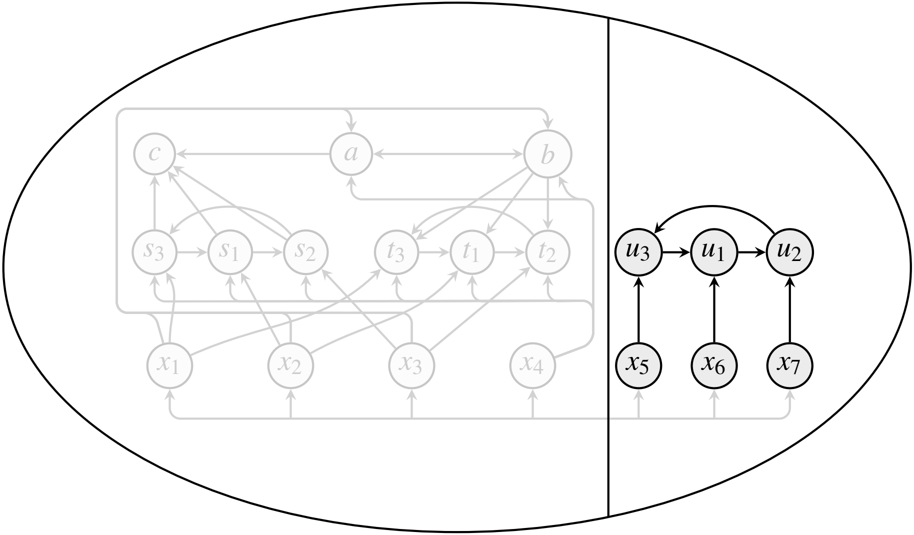
Input/Output AFs [Baroni et al., 2014]



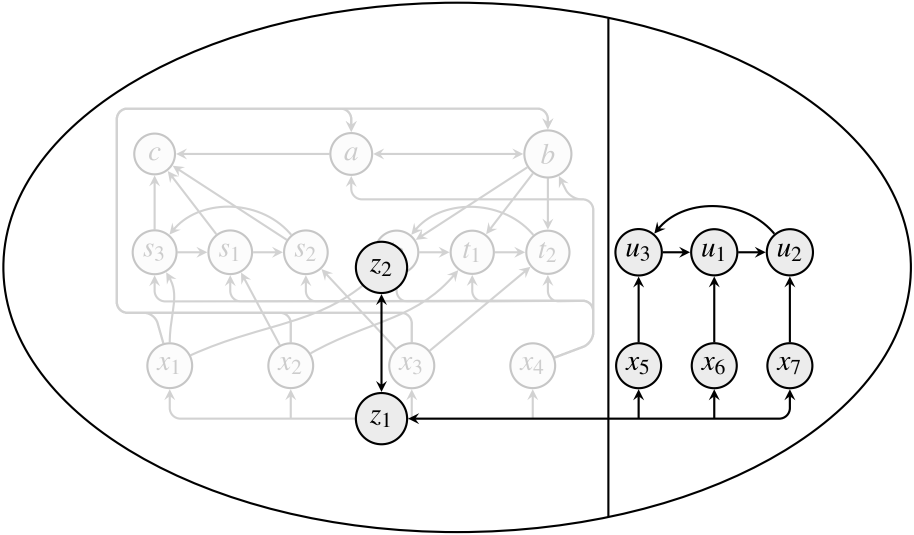
Input/Output AFs [Baroni et al., 2014]

- 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.

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.

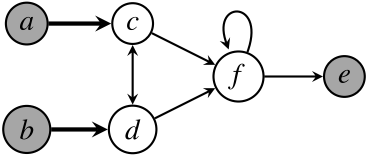
	stb	*prf*	*com*	*grd*	*sem*	*id*
Decomposability	Yes	No	Yes	No	No	No
SCC-Decomposability	Yes	Yes	Yes	Yes	No	No
Transparency	Yes	No*	Yes	Yes	No	No
SCC-Transparency	Yes	Yes	Yes	Yes	No	No

* Yes under additional mild conditions

Extension-based I/O-characterization

Definition

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 $I_F^- = \emptyset$.

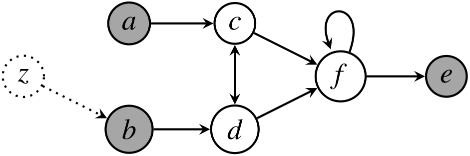


Extension-based I/O-characterization

Definition

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

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 $\mathfrak{p} : 2^I \mapsto 2^{2^O}$.

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

Extension-based I/O-characterization

Definition

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

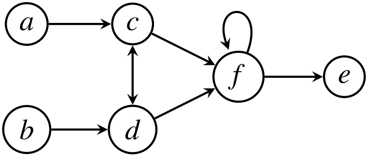
Input ($I = \{a, b\}$)	Output ($O = \{e\}$)
$\emptyset$	$\{\{e\}\}$
$\{a\}$	$\{\{e\}\}$
$\{b\}$	$\{\{e\}\}$
$\{a, b\}$	$\{\emptyset\}$

Definition

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

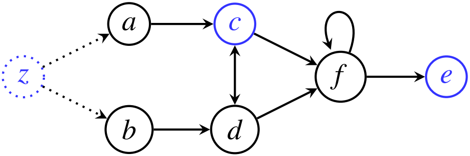
Extension-based I/O-characterization

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



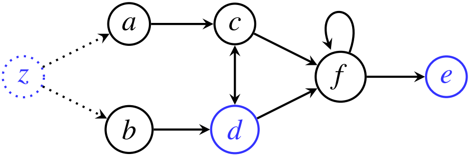
Extension-based I/O-characterization

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



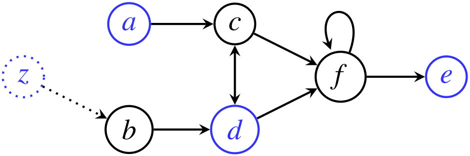
Extension-based I/O-characterization

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



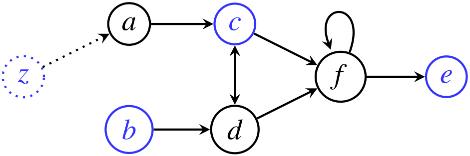
Extension-based I/O-characterization

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



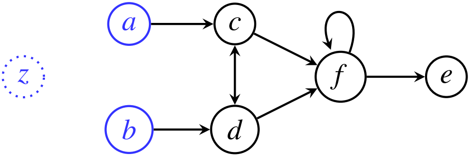
Extension-based I/O-characterization

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



Extension-based I/O-characterization

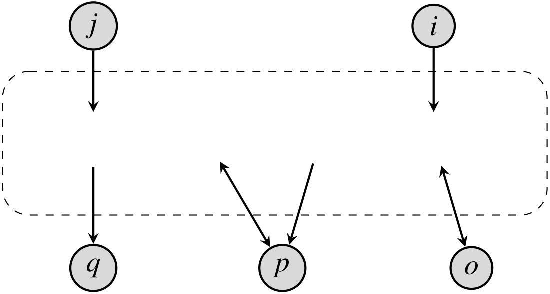
Input ($I = \{a, b\}$)	Output ($O = \{e\}$)
$\emptyset$	$\{\{e\}\}$
$\{a\}$	$\{\{e\}\}$
$\{b\}$	$\{\{e\}\}$
$\{a, b\}$	$\{\emptyset\}$



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

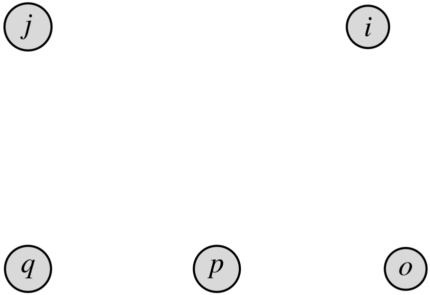
Theorem

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

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

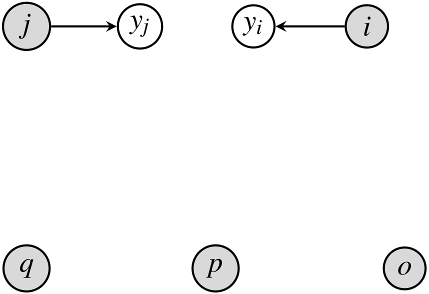
Extension-based I/O-characterization

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



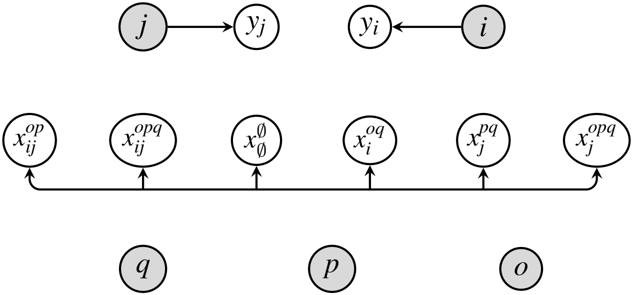
Extension-based I/O-characterization

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



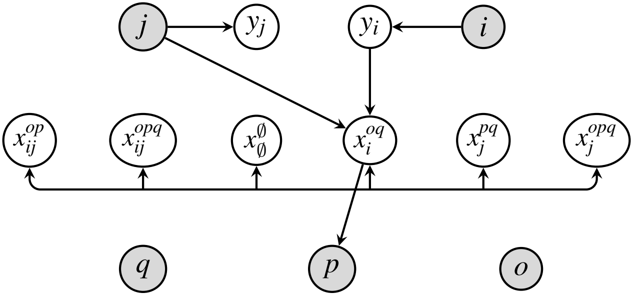
Extension-based I/O-characterization

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



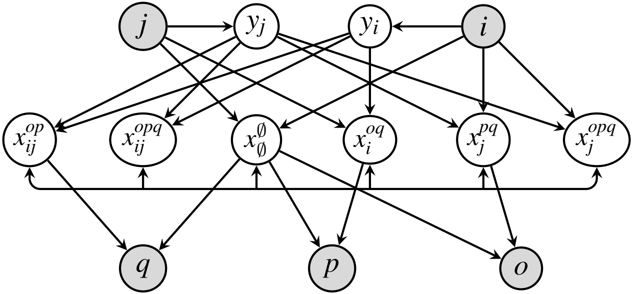
Extension-based I/O-characterization

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



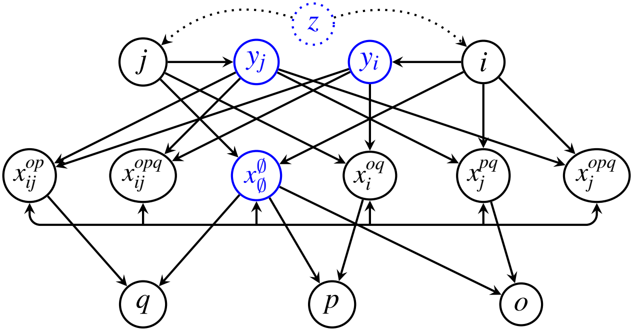
Extension-based I/O-characterization

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



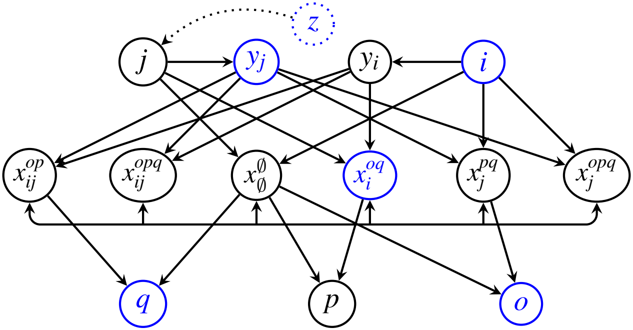
Extension-based I/O-characterization

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



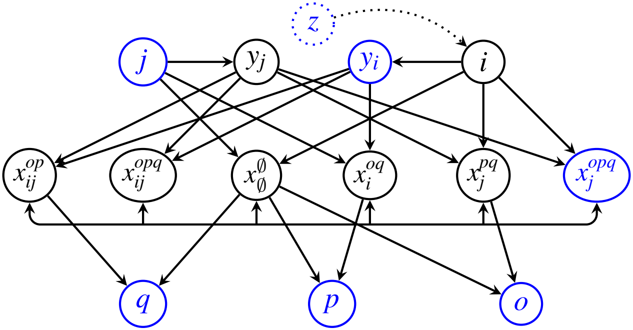
Extension-based I/O-characterization

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



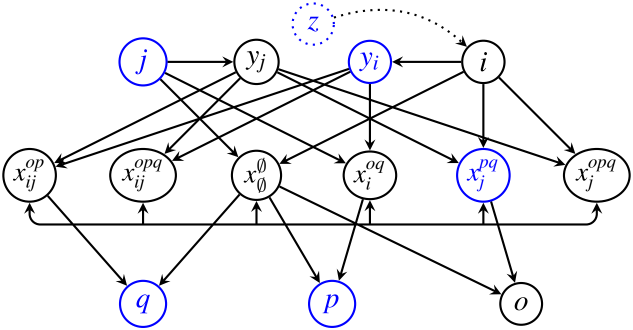
Extension-based I/O-characterization

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



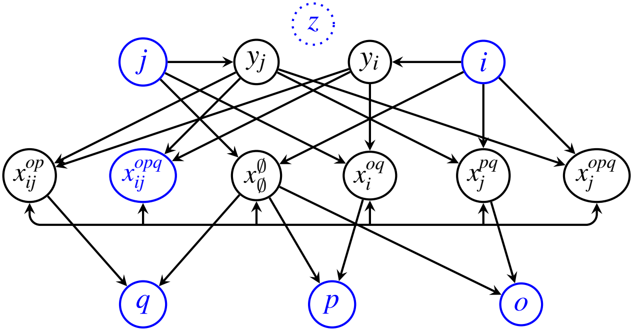
Extension-based I/O-characterization

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



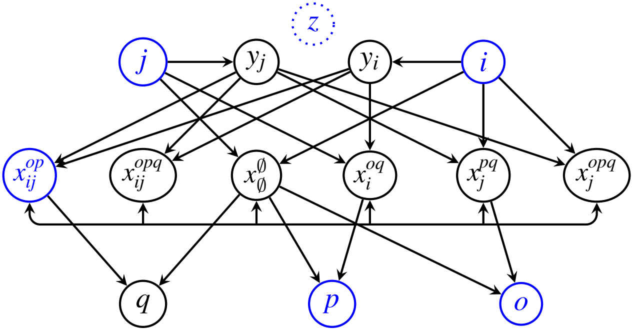
Extension-based I/O-characterization

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 ($I = \{i_1, i_2\}$)	Output ($O = \{o_1, o_2\}$)
$\{i_1 \leftarrow \mathtt{u}, i_2 \leftarrow \mathtt{u}\}$	$\{\{o_1 \leftarrow \mathtt{u}, o_2 \leftarrow \mathtt{u}\}\}$
$\{i_1 \leftarrow \mathtt{t}, i_2 \leftarrow \mathtt{u}\}$	$\{\{o_1 \leftarrow \mathtt{t}, o_2 \leftarrow \mathtt{u}\}\}$
$\{i_1 \leftarrow \mathtt{u}, i_2 \leftarrow \mathtt{t}\}$	$\{\{o_1 \leftarrow \mathtt{u}, o_2 \leftarrow \mathtt{t}\}\}$
$\{i_1 \leftarrow \mathtt{f}, i_2 \leftarrow \mathtt{u}\}$	$\{\{o_1 \leftarrow \mathtt{u}, o_2 \leftarrow \mathtt{u}\}\}$
$\{i_1 \leftarrow \mathtt{u}, i_2 \leftarrow \mathtt{f}\}$	$\{\{o_1 \leftarrow \mathtt{u}, o_2 \leftarrow \mathtt{f}\}\}$
$\{i_1 \leftarrow \mathtt{t}, i_2 \leftarrow \mathtt{t}\}$	$\{\{o_1 \leftarrow \mathtt{t}, o_2 \leftarrow \mathtt{f}\}\}$
$\{i_1 \leftarrow \mathtt{t}, i_2 \leftarrow \mathtt{f}\}$	$\{\{o_1 \leftarrow \mathtt{t}, o_2 \leftarrow \mathtt{f}\}\}$
$\{i_1 \leftarrow \mathtt{f}, i_2 \leftarrow \mathtt{t}\}$	$\{\{o_1 \leftarrow \mathtt{u}, o_2 \leftarrow \mathtt{t}\}\}$
$\{i_1 \leftarrow \mathtt{f}, i_2 \leftarrow \mathtt{f}\}$	$\{\{o_1 \leftarrow \mathtt{t}, o_2 \leftarrow \mathtt{f}\}\}$

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 ($I = \{i_1, i_2\}$)	Output ($O = \{o_1, o_2\}$)
$\{i_1 \leftarrow \mathtt{u}, i_2 \leftarrow \mathtt{u}\}$	$\{\{o_1 \leftarrow \mathtt{u}, o_2 \leftarrow \mathtt{u}\}\}$
$\{i_1 \leftarrow \mathtt{t}, i_2 \leftarrow \mathtt{u}\}$	$\{\{o_1 \leftarrow \mathtt{u}, o_2 \leftarrow \mathtt{u}\}, \{o_1 \leftarrow \mathtt{u}, o_2 \leftarrow \mathtt{f}\}\}$
$\vdots$	$\vdots$
$\{i_1 \leftarrow \mathtt{t}, i_2 \leftarrow \mathtt{t}\}$	$\{\{o_1 \leftarrow \mathtt{u}, o_2 \leftarrow \mathtt{u}\}\}$
$\{i_1 \leftarrow \mathtt{t}, i_2 \leftarrow \mathtt{f}\}$	$\{\{o_1 \leftarrow \mathtt{t}, o_2 \leftarrow \mathtt{f}\}, \{o_1 \leftarrow \mathtt{t}, o_2 \leftarrow \mathtt{t}\}\}$
$\vdots$	$\vdots$

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.

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.

Future Work

- 3-valued I/O-characterization of complete semantics.
- Partial I/O-specifications.
- 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

- 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.