Vorlesung “Formale Argumentation”

1. Einführung und Überblick

Ringo Baumann
Professur für Formale Argumentation
und Logisches Schließen

04. April 2024
Leipzig

Monotonic vs. Non-monotonic Logics

Monotonic vs. Non-monotonic **Logics**

- do not allow for a retraction of inferences, i.e.

$$\text{If } S \subseteq T, \text{ then } Cn(S) \subseteq Cn(T).$$

$$\text{If } S \subseteq T \text{ and } S \models \phi, \text{ then } T \models \phi.$$

- propositional logic, first-order logic, intuitionistic logic, ...

Monotonic vs. Non-monotonic **Logics**

- do not allow for a retraction of inferences, i.e.

$$\text{If } S \subseteq T, \text{ then } Cn(S) \subseteq Cn(T).$$

$$\text{If } S \subseteq T \text{ and } S \models \phi, \text{ then } T \models \phi.$$

- propositional logic, first-order logic, intuitionistic logic, ...
- monotonic reasoning is good for mathematics
- Example: group axioms, uniqueness of the neutral element

Monotonic vs. **Non-monotonic Logics**

- represent defeasible inference, i.e.

$$S \subseteq T \text{ and } Cn(S) \nsubseteq Cn(T) \text{ is possible.}$$

$$S \subseteq T,\ S \models \phi \text{ and } T \not\models \phi \text{ is possible.}$$

- default logic, circumscription, autoepistemic logic, . . .

Monotonic vs. **Non-monotonic Logics**

- represent defeasible inference, i.e.

$$S \subseteq T \text{ and } Cn(S) \nsubseteq Cn(T) \text{ is possible.}$$

$$S \subseteq T,\ S \models \phi \text{ and } T \not\models \phi \text{ is possible.}$$

- default logic, circumscription, autoepistemic logic, . . .
- reason: incomplete and/or uncertain information
- defeasible reasoning is the reasoning mode for “daily life”

☞ draw conclusions defeasibly

Monotonic vs. **Non-monotonic Logics**

Draw conclusions based on normality assumptions.

Monotonic vs. **Non-monotonic Logics**

Draw conclusions based on normality assumptions.

- Professors teach
- Birds fly
- Owls hunt at night
- Students don't like the 7th and 8th period
- Waiting for two hours at the doctor's office is frustrating
- The human heart is on the left side
- Kids like ice cream

Monotonic vs. **Non-monotonic Logics**

Draw conclusions based on normality assumptions.

- Professors teach . . . unless they are on sabbatical.
- Birds fly
- Owls hunt at night
- Students don't like the 7th and 8th period
- Waiting for two hours at the doctor's office is frustrating
- The human heart is on the left side
- Kids like ice cream

Monotonic vs. **Non-monotonic Logics**

Draw conclusions based on normality assumptions.

- Professors teach
- Birds fly . . . unless they are penguins.
- Owls hunt at night
- Students don't like the 7th and 8th period
- Waiting for two hours at the doctor's office is frustrating
- The human heart is on the left side
- Kids like ice cream

Monotonic vs. **Non-monotonic Logics**

Draw conclusions based on normality assumptions.

- Professors teach
- Birds fly
- Owls hunt at night . . . unless they live in a zoo.
- Students don't like the 7th and 8th period
- Waiting for two hours at the doctor's office is frustrating
- The human heart is on the left side
- Kids like ice cream

Monotonic vs. **Non-monotonic Logics**

Draw conclusions based on normality assumptions.

- Professors teach
- Birds fly
- Owls hunt at night
- Students don't like the 7th and 8th period . . . unless it's their favorite subject.
- Waiting for two hours at the doctor's office is frustrating
- The human heart is on the left side
- Kids like ice cream

Monotonic vs. **Non-monotonic Logics**

Draw conclusions based on normality assumptions.

- Professors teach
- Birds fly
- Owls hunt at night
- Students don't like the 7th and 8th period

- Waiting for two hours at the doctor's office is frustrating . . . unless you are close to finish a proof.
- The human heart is on the left side

- Kids like ice cream

Monotonic vs. **Non-monotonic Logics**

Draw conclusions based on normality assumptions.

- Professors teach
- Birds fly
- Owls hunt at night
- Students don't like the 7th and 8th period
- Waiting for two hours at the doctor's office is frustrating
- The human heart is on the left side . . . unless one has dextrocardia.
- Kids like ice cream

Monotonic vs. **Non-monotonic Logics**

Draw conclusions based on normality assumptions.

- Professors teach
- Birds fly
- Owls hunt at night
- Students don't like the 7th and 8th period
- Waiting for two hours at the doctor's office is frustrating
- The human heart is on the left side
- Kids like ice cream . . . ~~unless~~ no exceptions!

Non-monotonic Logics

Example (Rule-based Formalism)

1. Knowledge Base

$$r_1: \quad \Rightarrow a$$
$$r_2: \quad a \Rightarrow b$$
$$r_3: \quad b \rightarrow not\ a$$
$$r_4: \quad \rightarrow c$$
$$r_5: \quad c \Rightarrow not\ b$$

If a, then normally b.

If b, then definitely not a.

Example (Rule-based Formalism)

1. Knowledge Base

$r_1 : \quad \Rightarrow a$

$r_2 : \quad a \Rightarrow b$

$r_3 : \quad b \rightarrow not\ a$

$r_4 : \quad \rightarrow c$

$r_5 : \quad c \Rightarrow not\ b$

If a, then normally b.

If b, then definitely not a.

2. Conclusion

Conc = {a,c}

Towards Abstract Argumentation - The Paradigm Shift

Seminal Paper by Phan Minh Dung,
On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n- person games, AIJ, 1995.

Towards Abstract Argumentation - The Paradigm Shift

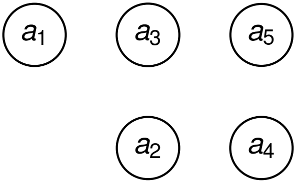
Seminal Paper by Phan Minh Dung,
On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n- person games, AIJ, 1995.

Two main ideas:

1. non-monotonic reasoning can be modelled as a kind of argumentation
2. determining the acceptability of arguments can be done on an abstract level

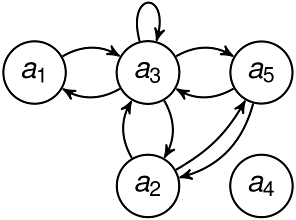
Abstract away from

- the internal structure of arguments, and (nodes)



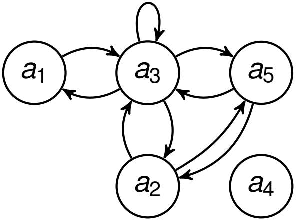
Abstract away from

- the internal structure of arguments, and (nodes)
- the reason why an argument attacks an other (edges)



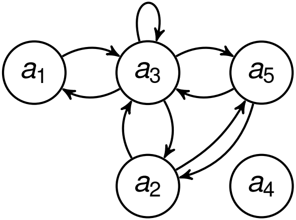
Abstract away from

- the internal structure of arguments, and (nodes)
- the reason why an argument attacks an other (edges)

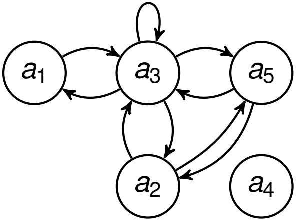


☞ an argumentation scenario is simply a directed graphs

How to select reasonable positions?



How to select reasonable positions?



Definition

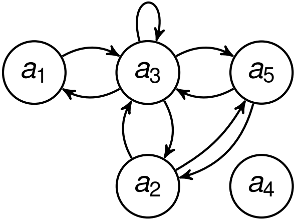
A semantics is a total function

$$\sigma : \mathcal{F} \to 2^{2^{\mathcal{U}}} \quad F = (A, R) \mapsto \sigma(F) \subseteq 2^{A}.$$

($\mathcal{F}$ - set of all AFs)

($\mathcal{U}$ - set of all arguments)

Semantics (select reasonable positions)



$ad(F) = \{\emptyset, \{a_1\}, \{a_2\},$
$\{a_4\}, \{a_5\}, \{a_1, a_2\},$
$\{a_1, a_4\}, \{a_1, a_5\},$
$\{a_2, a_4\}, \{a_4, a_5\},$
$\{a_1, a_2, a_4\}, \{a_1, a_4, a_5\}\}$

Definition

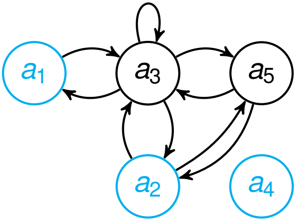
Admissible semantics is a total function

$$ad : \mathcal{F} \to 2^{2^{\mathcal{U}}} \quad F = (A, R) \mapsto ad(F) \subseteq 2^A.$$

$E \in ad(F)$ iff

1. $\forall a, b \in E : (a, b) \notin R$ (conflict-freeness)
2. $\forall a, b\ ((a, b) \in R \land b \in E \to \exists c \in E : (c, a) \in R)$ (defense)

Semantics (select reasonable positions)



$ad(F) = \{\emptyset, \{a_1\}, \{a_2\}, \{a_4\}, \{a_5\}, \{a_1, a_2\}, \{a_1, a_4\}, \{a_1, a_5\}, \{a_2, a_4\}, \{a_4, a_5\}, \{a_1, a_2, a_4\}, \{a_1, a_4, a_5\}\}$

Definition

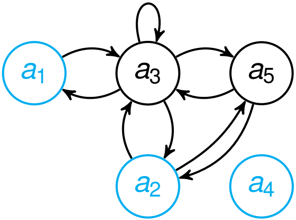
Admissible semantics is a total function

$$ad : \mathcal{F} \to 2^{2^{\mathcal{U}}} \quad F = (A, R) \mapsto ad(F) \subseteq 2^A.$$

$E \in ad(F)$ iff

1. $\forall a, b \in E : (a, b) \notin R$ (conflict-freeness)
2. $\forall a, b\ ((a, b) \in R \land b \in E \to \exists c \in E : (c, a) \in R)$ (defense)

Semantics (select reasonable positions)



$ad(F) = \{\emptyset, \{a_1\}, \{a_2\}, \{a_4\}, \{a_5\}, \{a_1, a_2\}, \{a_1, a_4\}, \{a_1, a_5\}, \{a_2, a_4\}, \{a_4, a_5\}, \{a_1, a_2, a_4\}, \{a_1, a_4, a_5\}\}$

Definition

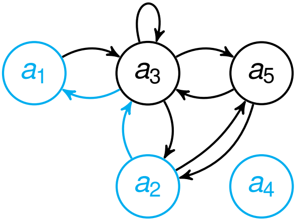
Admissible semantics is a total function

$$ad : \mathcal{F} \to 2^{2^{\mathcal{U}}} \quad F = (A, R) \mapsto ad(F) \subseteq 2^A.$$

$E \in ad(F)$ iff

1. $\forall a, b \in E : (a, b) \notin R$ ✔ (conflict-freeness)
2. $\forall a, b \, ((a, b) \in R \wedge b \in E \to \exists c \in E : (c, a) \in R)$ (defense)

Semantics (select reasonable positions)



$ad(F) = \{\emptyset, \{a_1\}, \{a_2\},$
$\{a_4\}, \{a_5\}, \{a_1, a_2\},$
$\{a_1, a_4\}, \{a_1, a_5\},$
$\{a_2, a_4\}, \{a_4, a_5\},$
$\{a_1, a_2, a_4\}, \{a_1, a_4, a_5\}\}$

Definition

Admissible semantics is a total function

$$ad : \mathcal{F} \to 2^{2^{\mathcal{U}}} \quad F = (A, R) \mapsto ad(F) \subseteq 2^A.$$

$E \in ad(F)$ iff

1. $\forall a, b \in E : (a, b) \notin R$ (conflict-freeness)
2. $\forall a, b\ ((a, b) \in R \wedge b \in E \to \exists c \in E : (c, a) \in R)$ ✔ (defense)

Reconstruction via Argumentation

Example (Rule-based Formalism)

1. Knowledge Base

$$
\begin{aligned}
r_1 &: \quad \Rightarrow a \\
r_2 &: \quad a \Rightarrow b \\
r_3 &: \quad b \rightarrow not\ a \\
r_4 &: \quad \rightarrow c \\
r_5 &: \quad c \Rightarrow not\ b
\end{aligned}
$$

Reconstruction via Argumentation

Example (Rule-based Formalism)

1. Knowledge Base

$r_1: \Rightarrow a$

$r_2: a \Rightarrow b$

$r_3: b \rightarrow not\ a$

$r_4: \rightarrow c$

$r_5: c \Rightarrow not\ b$

$\Longrightarrow$

2. Arguments

$a_1: [r_1 \mid a]$

$a_2: [a_1, r_2 \mid b]$

$a_3: [a_2, r_3 \mid not\ a]$

$a_4: [r_4 \mid c]$

$a_5: [a_4, r_5 \mid not\ b]$

Reconstruction via Argumentation

Example (Rule-based Formalism)

1. Knowledge Base

$r_1 : \;\; \Rightarrow a$

$r_2 : \;\; a \Rightarrow b$

$r_3 : \;\; b \rightarrow not\ a$

$r_4 : \;\; \rightarrow c$

$r_5 : \;\; c \Rightarrow not\ b$

$\Longrightarrow$

2. Arguments

$a_1 : \;\; [r_1 \mid a]$

$a_2 : \;\; [a_1, r_2 \mid b]$

$a_3 : \;\; [a_2, r_3 \mid not\ a]$

$a_4 : \;\; [r_4 \mid c]$

$a_5 : \;\; [a_4, r_5 \mid not\ b]$

a_1 claims a justified by r_1

a_2 claims b justified by a_1 and r_2

Reconstruction via Argumentation

Example (Rule-based Formalism)

1. Knowledge Base	2. Arguments	3. Conflicts
$r_1: \Rightarrow a$	$a_1: [r_1 \mid a]$	c_1 : a_1 attacks a_3
$r_2: a \Rightarrow b$	$a_2: [a_1, r_2 \mid b]$	c_2, c_3 : a_2 attacks a_3, a_5
$r_3: b \rightarrow \textit{not } a$	$a_3: [a_2, r_3 \mid \textit{not } a]$	c_4, c_5, c_6, c_7 : a_3 attacks a_1, a_2, a_3, a_5
$r_4: \rightarrow c$	$a_4: [r_4 \mid c]$	a_4 attacks no-one
$r_5: c \Rightarrow \textit{not } b$	$a_5: [a_4, r_5 \mid \textit{not } b]$	c_8, c_9 : a_5 attacks a_2, a_3

Reconstruction via Argumentation

Example (Rule-based Formalism)

1. Knowledge Base

r_1 :	$\Rightarrow a$
r_2 :	$a \Rightarrow b$
r_3 :	$b \rightarrow not\ a$
r_4 :	$\rightarrow c$
r_5 :	$c \Rightarrow not\ b$

2. Arguments

a_1 :	$[r_1 \mid a]$
a_2 :	$[a_1, r_2 \mid b]$
a_3 :	$[a_2, r_3 \mid not\ a]$
a_4 :	$[r_4 \mid c]$
a_5 :	$[a_4, r_5 \mid not\ b]$

3. Conflicts

c_1 :	a_1 attacks a_3
c_2, c_3 :	a_2 attacks a_3, a_5
c_4, c_5, c_6, c_7 :	a_3 attacks a_1, a_2, a_3, a_5
	a_4 attacks no-one
c_8, c_9 :	a_5 attacks a_2, a_3

Reconstruction via Argumentation

Example (Rule-based Formalism)

1. Knowledge Base

r_1 :	$\Rightarrow a$
r_2 :	$a \Rightarrow b$
r_3 :	$b \rightarrow \textit{not}\ a$
r_4 :	$\rightarrow c$
r_5 :	$c \Rightarrow \textit{not}\ b$

⇒

2. Arguments

a_1 :	$[r_1 \mid a]$
a_2 :	$[a_1, r_2 \mid b]$
a_3 :	$[a_2, r_3 \mid \textit{not}\ a]$
a_4 :	$[r_4 \mid c]$
a_5 :	$[a_4, r_5 \mid \textit{not}\ b]$

⇒

3. Conflicts

c_1 :	a_1 attacks a_3
c_2, c_3 :	a_2 attacks a_3, a_5
c_4, c_5, c_6, c_7 :	a_3 attacks a_1, a_2, a_3, a_5
	a_4 attacks no-one
c_8, c_9 :	a_5 attacks a_2, a_3

Reconstruction via Argumentation

Example (Rule-based Formalism)

1. Knowledge Base

r_1 :	$\Rightarrow a$
r_2 :	$a \Rightarrow b$
r_3 :	$b \rightarrow \mathit{not}\ a$
r_4 :	$\rightarrow c$
r_5 :	$c \Rightarrow \mathit{not}\ b$

2. Arguments

a_1 :	$[r_1 \mid a]$
a_2 :	$[a_1, r_2 \mid b]$
a_3 :	$[a_2, r_3 \mid \mathit{not}\ a]$
a_4 :	$[r_4 \mid c]$
a_5 :	$[a_4, r_5 \mid \mathit{not}\ b]$

3. Conflicts

c_1 :	a_1 attacks a_3
c_2, c_3 :	a_2 attacks a_3, a_5
c_4, c_5, c_6, c_7 :	a_3 attacks a_1, a_2, a_3, a_5
	a_4 attacks no-one
c_8, c_9 :	a_5 attacks a_2, a_3

Reconstruction via Argumentation

Example (Rule-based Formalism)

1. Knowledge Base

$r_1 : \;\; \Rightarrow a$

$r_2 : \;\; a \Rightarrow b$

$r_3 : \;\; b \rightarrow \text{not } a$

$r_4 : \;\; \rightarrow c$

$r_5 : \;\; c \Rightarrow \text{not } b$

2. Arguments

$a_1 : \;\; [r_1 \mid a]$

$a_2 : \;\; [a_1, r_2 \mid b]$

$a_3 : \;\; [a_2, r_3 \mid \text{not } a]$

$a_4 : \;\; [r_4 \mid c]$

$a_5 : \;\; [a_4, r_5 \mid \text{not } b]$

3. Conflicts

$c_1 :$ a_1 attacks a_3

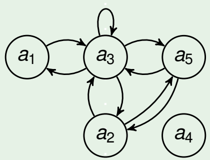
$c_2, c_3 :$ a_2 attacks a_3, a_5

$c_4, c_5, c_6, c_7 :$ a_3 attacks a_1, a_2, a_3, a_5

a_4 attacks no-one

$c_8, c_9 :$ a_5 attacks a_2, a_3

4. Instantiation



Reconstruction via Argumentation

Example (Rule-based Formalism)

1. Knowledge Base

$r_1 : \quad \Rightarrow a$

$r_2 : \quad a \Rightarrow b$

$r_3 : \quad b \rightarrow not\ a$

$r_4 : \quad \rightarrow c$

$r_5 : \quad c \Rightarrow not\ b$

2. Arguments

$a_1 : \quad [r_1 \mid a]$

$a_2 : \quad [a_1, r_2 \mid b]$

$a_3 : \quad [a_2, r_3 \mid not\ a]$

$a_4 : \quad [r_4 \mid c]$

$a_5 : \quad [a_4, r_5 \mid not\ b]$

3. Conflicts

$c_1 :$ a_1 attacks a_3

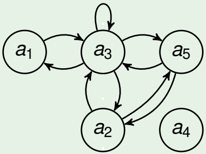
$c_2, c_3 :$ a_2 attacks a_3, a_5

$c_4, c_5, c_6, c_7 :$ a_3 attacks a_1, a_2, a_3, a_5

a_4 attacks no-one

$c_8, c_9 :$ a_5 attacks a_2, a_3

4. Instantiation



5. Resolving

$E_1 = \{a_1, a_2, a_4\}$

$E_2 = \{a_1, a_4, a_5\}$

Reconstruction via Argumentation

Example (Rule-based Formalism)

1. Knowledge Base

$r_1 : \quad \Rightarrow a$

$r_2 : \quad a \Rightarrow b$

$r_3 : \quad b \rightarrow not\ a$

$r_4 : \quad \rightarrow c$

$r_5 : \quad c \Rightarrow not\ b$

2. Arguments

$a_1 : \quad [r_1 \mid a]$

$a_2 : \quad [a_1, r_2 \mid b]$

$a_3 : \quad [a_2, r_3 \mid not\ a]$

$a_4 : \quad [r_4 \mid c]$

$a_5 : \quad [a_4, r_5 \mid not\ b]$

3. Conflicts

$c_1 :$ a_1 attacks a_3

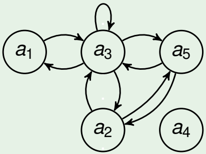
$c_2, c_3 :$ a_2 attacks a_3, a_5

$c_4, c_5, c_6, c_7 :$ a_3 attacks a_1, a_2, a_3, a_5

a_4 attacks no-one

$c_8, c_9 :$ a_5 attacks a_2, a_3

4. Instantiation



5. Resolving

$E_1 = \{a_1, a_2, a_4\}$

$E_2 = \{a_1, a_4, a_5\}$

6. Conclusion

$E_1 = \{a, b, c\}$

$E_2 = \{a, c, not\ b\}$

Conc = {a,c}

Example (Rule-based Formalism)

1. Knowledge Base

$r_1: \;\; \Rightarrow a$

$r_2: \;\; a \Rightarrow b$

$r_3: \;\; b \rightarrow not\ a$

$r_4: \;\; \rightarrow c$

$r_5: \;\; c \Rightarrow not\ b$

2. Arguments

$a_1: \;\; [r_1 \mid a]$

$a_2: \;\; [a_1, r_2 \mid b]$

$a_3: \;\; [a_2, r_3 \mid not\ a]$

$a_4: \;\; [r_4 \mid c]$

$a_5: \;\; [a_4, r_5 \mid not\ b]$

3. Conflicts

$c_1:$ a_1 attacks a_3

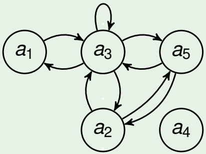
$c_2, c_3:$ a_2 attacks a_3, a_5

$c_4, c_5, c_6, c_7:$ a_3 attacks a_1, a_2, a_3, a_5

a_4 attacks no-one

$c_8, c_9:$ a_5 attacks a_2, a_3

4. Instantiation



5. Resolving

$E_1 = \{a_1, a_2, a_4\}$

$E_2 = \{a_1, a_4, a_5\}$

6. Conclusion

$E_1 = \{a, b, c\}$

$E_2 = \{a, c, not\ b\}$

Conc = {a,c}

Explainability

EU's General Data Protection Regulation, 2018

" ...establishes a right for all individuals to obtain meaningful explanations of the logic involved when automated (algorithmic) decision making takes place."

Explainability

EU's General Data Protection Regulation, 2018

" ...establishes a right for all individuals to obtain meaningful explanations of the logic involved when automated (algorithmic) decision making takes place."

German AI strategy, 2020

"...making AI explainable, accountable, and transparent is the key to winning over the public's trust. "

Explainability

EU's General Data Protection Regulation, 2018

" ...establishes a right for all individuals to obtain meaningful explanations of the logic involved when automated (algorithmic) decision making takes place."

German AI strategy, 2020

"...making AI explainable, accountable, and transparent is the key to winning over the public's trust. "

EU Artificial Intelligence Act, March 2024

"...aims to classify and regulate AI applications based on their risk to cause harm."

Reconstruction, Explanation, Semantics via Argumentation

Example (Rule-based Formalism)

1. Knowledge Base

$r_1 : \quad \Rightarrow a$

$r_2 : \quad a \Rightarrow b$

$r_3 : \quad b \rightarrow \textit{not } a$

$r_4 : \quad \rightarrow c$

$r_5 : \quad c \Rightarrow \textit{not } b$

2. Arguments

$a_1 : \quad [r_1 \mid a]$

$a_2 : \quad [a_1, r_2 \mid b]$

$a_3 : \quad [a_2, r_3 \mid \textit{not } a]$

$a_4 : \quad [r_4 \mid c]$

$a_5 : \quad [a_4, r_5 \mid \textit{not } b]$

3. Conflicts

$c_1 :$ a_1 attacks a_3

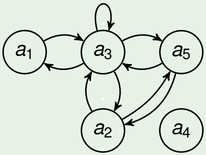
$c_2, c_3 :$ a_2 attacks a_3, a_5

$c_4, c_5, c_6, c_7 :$ a_3 attacks a_1, a_2, a_3, a_5

a_4 attacks no-one

$c_8, c_9 :$ a_5 attacks a_2, a_3

4. Instantiation



5. Resolving

$E_1 = \{a_1, a_2, a_4\}$

$E_2 = \{a_1, a_4, a_5\}$

6. Conclusion

$E_1 = \{a, b, c\}$

$E_2 = \{a, c, \textit{not } b\}$

Conc = {a,c}

Some Issues:

Some Issues: Simplification

Example (Propositional Logic)

$$S = \{a, a \to b, \neg b \vee c, e \wedge f \to d, d \leftrightarrow e\}$$

Some Issues: Simplification

Example (Propositional Logic)

$$S = \{a, a \to b, \neg b \vee c, e \wedge f \to d, d \leftrightarrow e\}$$

$$\equiv \{a, \top \to b, \neg b \vee c, e \wedge f \to d, d \leftrightarrow e\}$$

Some Issues: Simplification

Example (Propositional Logic)

$$
\begin{aligned}
S &= \{a, a \to b, \neg b \lor c, e \land f \to d, d \leftrightarrow e\} \\
&\equiv \{a, \top \to b, \neg b \lor c, e \land f \to d, d \leftrightarrow e\} \\
&\equiv \{a, b \qquad , \neg b \lor c, e \land f \to d, d \leftrightarrow e\}
\end{aligned}
$$

Some Issues: Simplification

Example (Propositional Logic)

$$
\begin{aligned}
S &= \{a, a \to b, \neg b \lor c, e \land f \to d, d \leftrightarrow e\} \\
&\equiv \{a, \top \to b, \neg b \lor c, e \land f \to d, d \leftrightarrow e\} \\
&\equiv \{a, b \qquad , \neg b \lor c, e \land f \to d, d \leftrightarrow e\} \\
&\equiv \{a, b \qquad , \neg \top \lor c, e \land f \to d, d \leftrightarrow e\}
\end{aligned}
$$

Some Issues: Simplification

Example (Propositional Logic)

$$
\begin{aligned}
S &= \{a, a \to b, \neg b \lor c, e \land f \to d, d \leftrightarrow e\} \\
&\equiv \{a, \top \to b, \neg b \lor c, e \land f \to d, d \leftrightarrow e\} \\
&\equiv \{a, b \quad , \neg b \lor c, e \land f \to d, d \leftrightarrow e\} \\
&\equiv \{a, b \quad , \neg \top \lor c, e \land f \to d, d \leftrightarrow e\} \\
&\equiv \{a, b \quad , \quad \bot \lor c, e \land f \to d, d \leftrightarrow e\}
\end{aligned}
$$

Some Issues: Simplification

Example (Propositional Logic)

$$
\begin{aligned}
S &= \{a, a \to b, \neg b \lor c, e \land f \to d, d \leftrightarrow e\} \\
&\equiv \{a, \top \to b, \neg b \lor c, e \land f \to d, d \leftrightarrow e\} \\
&\equiv \{a, b, \neg b \lor c, e \land f \to d, d \leftrightarrow e\} \\
&\equiv \{a, b, \neg \top \lor c, e \land f \to d, d \leftrightarrow e\} \\
&\equiv \{a, b, \bot \lor c, e \land f \to d, d \leftrightarrow e\} \\
&\equiv \{a, b, c, e \land f \to d, d \leftrightarrow e\}
\end{aligned}
$$

Some Issues: Simplification

Example (Propositional Logic)

$$
\begin{aligned}
S &= \{a, a \rightarrow b, \neg b \vee c, e \wedge f \rightarrow d, d \leftrightarrow e\} \\
&\equiv \{a, \top \rightarrow b, \neg b \vee c, e \wedge f \rightarrow d, d \leftrightarrow e\} \\
&\equiv \{a, b \qquad , \neg b \vee c, e \wedge f \rightarrow d, d \leftrightarrow e\} \\
&\equiv \{a, b \qquad , \neg \top \vee c, e \wedge f \rightarrow d, d \leftrightarrow e\} \\
&\equiv \{a, b \qquad , \quad \bot \vee c, e \wedge f \rightarrow d, d \leftrightarrow e\} \\
&\equiv \{a, b \qquad , c \qquad , e \wedge f \rightarrow d, d \leftrightarrow e\} \\
&\equiv \{a, b \qquad , c \qquad , d \wedge f \rightarrow d, d \leftrightarrow e\}
\end{aligned}
$$

Some Issues: Simplification

Example (Propositional Logic)

$$
\begin{aligned}
S = \{&a, a \to b, \neg b \lor c, e \land f \to d, d \leftrightarrow e\} \\
\equiv \{&a, \top \to b, \neg b \lor c, e \land f \to d, d \leftrightarrow e\} \\
\equiv \{&a, b, \neg b \lor c, e \land f \to d, d \leftrightarrow e\} \\
\equiv \{&a, b, \neg \top \lor c, e \land f \to d, d \leftrightarrow e\} \\
\equiv \{&a, b, \bot \lor c, e \land f \to d, d \leftrightarrow e\} \\
\equiv \{&a, b, c, e \land f \to d, d \leftrightarrow e\} \\
\equiv \{&a, b, c, d \land f \to d, d \leftrightarrow e\} \\
\equiv \{&a, b, c, \top, d \leftrightarrow e\}
\end{aligned}
$$

Some Issues: Simplification

Example (Propositional Logic)

$$
\begin{aligned}
S &= \{a, a \to b, \neg b \vee c, e \wedge f \to d, d \leftrightarrow e\} \\
&\equiv \{a, \top \to b, \neg b \vee c, e \wedge f \to d, d \leftrightarrow e\} \\
&\equiv \{a, b, \neg b \vee c, e \wedge f \to d, d \leftrightarrow e\} \\
&\equiv \{a, b, \neg \top \vee c, e \wedge f \to d, d \leftrightarrow e\} \\
&\equiv \{a, b, \bot \vee c, e \wedge f \to d, d \leftrightarrow e\} \\
&\equiv \{a, b, c, e \wedge f \to d, d \leftrightarrow e\} \\
&\equiv \{a, b, c, d \wedge f \to d, d \leftrightarrow e\} \\
&\equiv \{a, b, c, \top, d \leftrightarrow e\} \\
&\equiv \{a, b, c, d \leftrightarrow e\} = T
\end{aligned}
$$

Some Issues: Simplification

Example (Propositional Logic)

$$S = \{a, a \to b, \neg b \vee c, e \wedge f \to d, d \leftrightarrow e\} \text{ and}$$
$$T = \{a, b, c, d \leftrightarrow e\}$$

are equivalent, i.e. $Mod(S) = Mod(T)$.

Some Issues: Simplification

Example (Propositional Logic)

$$S = \{a, a \rightarrow b, \neg b \vee c, e \wedge f \rightarrow d, d \leftrightarrow e\} \text{ and}$$
$$T = \{a, b, c, d \leftrightarrow e\}$$

are equivalent, i.e. $Mod(S) = Mod(T)$.
Moreover, they are even strongly equivalent, i.e.

For each H, we have: $Mod(S \cup H) = Mod(T \cup H)$.

Proof:
$$\begin{aligned} Mod(S \cup H) &= Mod(S) \cap Mod(H) \\ &= Mod(T) \cap Mod(H) \\ &= Mod(T \cup H) \end{aligned}$$

Some Issues: Simplification

- Argumentation semantics σ does not possess the intersection property, i.e.

$$\sigma(F \sqcup H) \neq \sigma(F) \cap \sigma(H) \text{ is possible.}$$

Some Issues: Simplification

- Argumentation semantics σ does not possess the intersection property, i.e.

$$\sigma(F \sqcup H) \neq \sigma(F) \cap \sigma(H) \text{ is possible.}$$

- but, so-called kernels guarantee strong equivalence
- admissible kernel deletes an attack $(a, b) \in R$ if

$$a \neq b,\ (a, a) \in R,\ \{(b, a), (b, b)\} \cap R \neq \emptyset$$

Some Issues: Simplification

Example (Rule-based Formalism,)

1. Knowledge Base

$r_1 : \quad \Rightarrow a$

$r_2 : \quad a \Rightarrow b$

$r_3 : \quad b \rightarrow not\ a$

$r_4 : \quad \rightarrow c$

$r_5 : \quad c \Rightarrow not\ b$

2. Arguments

$a_1 : \quad [r_1 \mid a]$

$a_2 : \quad [a_1, r_2 \mid b]$

$a_3 : \quad [a_2, r_3 \mid not\ a]$

$a_4 : \quad [r_4 \mid c]$

$a_5 : \quad [a_4, r_5 \mid not\ b]$

3. Conflicts

$c_1 :$ a_1 attacks a_3

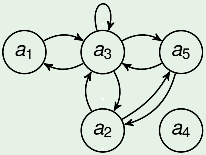
$c_2, c_3 :$ a_2 attacks a_3, a_5

$c_4, c_5, c_6, c_7 :$ a_3 attacks a_1, a_2, a_3, a_5

a_4 attacks no-one

$c_8, c_9 :$ a_5 attacks a_2, a_3

4. Instantiation



Some Issues: Simplification

Example (Rule-based Formalism, strong equivalence)

1. Knowledge Base

$r_1 : \quad \Rightarrow a$

$r_2 : \quad a \Rightarrow b$

$r_3 : \quad b \rightarrow \textit{not } a$

$r_4 : \quad \rightarrow c$

$r_5 : \quad c \Rightarrow \textit{not } b$

2. Arguments

$a_1 : \quad [r_1 \mid a]$

$a_2 : \quad [a_1, r_2 \mid b]$

$a_3 : \quad [a_2, r_3 \mid \textit{not } a]$

$a_4 : \quad [r_4 \mid c]$

$a_5 : \quad [a_4, r_5 \mid \textit{not } b]$

3. Conflicts

$c_1 :$ a_1 attacks a_3

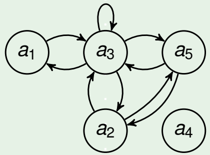
$c_2, c_3 :$ a_2 attacks a_3, a_5

$c_4, c_5, c_6, c_7 :$ a_3 attacks a_1, a_2, a_3, a_5

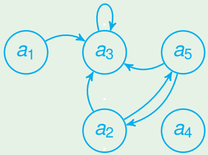
a_4 attacks no-one

$c_8, c_9 :$ a_5 attacks a_2, a_3

4. Instantiation



5. Simplification



Some Issues: Simplification

Example (strong expansion equivalence)

1. Knowledge Base

$r_1 : \Rightarrow a$
$r_2 : a \Rightarrow b$
$r_3 : b \rightarrow \text{not } a$
$r_4 : \rightarrow c$
$r_5 : c \Rightarrow \text{not } b$

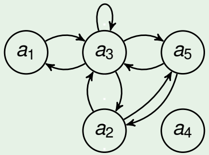
2. Arguments

$a_1 : [r_1 \mid a]$
$a_2 : [a_1, r_2 \mid b]$
$a_3 : [a_2, r_3 \mid \text{not } a]$
$a_4 : [r_4 \mid c]$
$a_5 : [a_4, r_5 \mid \text{not } b]$

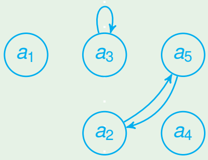
3. Conflicts

$c_1 :$ a_1 attacks a_3
$c_2, c_3 :$ a_2 attacks a_3, a_5
$c_4, c_5, c_6, c_7 :$ a_3 attacks a_1, a_2, a_3, a_5
a_4 attacks no-one
$c_8, c_9 :$ a_5 attacks a_2, a_3

4. Instantiation



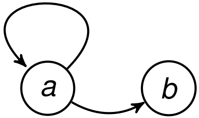
5. Simplification



Some Issues: Odd-cycles

- A 25 year old problem

"An interesting topic of research is the problem of self-defeating arguments as illustrated in the following example.



The only admissible extension here is empty though one can argue that since a defeats itself, b should be acceptable."

[Dung, 1995]

Some Issues: Odd-cycles

Definition

Weak Admissibility semantics is a total function

$$ad^w : \mathcal{F} \to 2^{2^{\mathcal{U}}} \quad F = (A, R) \mapsto ad^w(F) \subseteq 2^A.$$

$E \in ad^w(F)$ iff

Some Issues: Odd-cycles

Definition

Weak Admissibility semantics is a total function

$$ad^w : \mathcal{F} \to 2^{2^{\mathcal{U}}} \quad F = (A, R) \mapsto ad^w(F) \subseteq 2^A.$$

$E \in ad^w(F)$ iff

1. E is conflict-free, and
2. for any attacker y of E we have $y \notin \bigcup ad^w(F^E)$.

F^E is the AF F restricted to $A \setminus (E \cup E^+)$ (E-reduct)

KR 2020, *Comparing Weak Admissibility Semantics to their Dung-style Counterparts - Reduct, Modularization and Strong Equivalence*, R. Baumann, G. Brewka and M. Ulbricht (**Ray Reiter Best Paper Award**)

Some Issues: Odd-cycles

Definition

Weak Admissibility semantics is a total function

$$ad^w : \mathcal{F} \to 2^{2^{\mathcal{U}}} \quad F = (A, R) \mapsto ad^w(F) \subseteq 2^A.$$

$E \in ad^w(F)$ iff

1. E is conflict-free, and
2. for any attacker y of E we have $y \notin \bigcup ad^w(F^E)$.

F^E is the AF F restricted to $A \setminus (E \cup E^+)$ (E-reduct)

☞ recursive definition

KR 2020, *Comparing Weak Admissibility Semantics to their Dung-style Counterparts - Reduct, Modularization and Strong Equivalence*, R. Baumann, G. Brewka and M. Ulbricht (**Ray Reiter Best Paper Award**)

Some Issues: Odd-cycles

Definition

Weak Admissibility semantics is a total function

$$ad^w : \mathcal{F} \to 2^{2^{\mathcal{U}}} \quad F = (A, R) \mapsto ad^w(F) \subseteq 2^A.$$

$E \in ad^w(F)$ iff

1. E is conflict-free, and
2. any attacker y is counter-attacked or itself not acceptable

F^E is the AF F restricted to $A \setminus (E \cup E^+)$ (E-reduct)

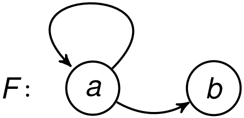
☞ main idea

KR 2020, *Comparing Weak Admissibility Semantics to their Dung-style Counterparts - Reduct, Modularization and Strong Equivalence*, R. Baumann, G. Brewka and M. Ulbricht (**Ray Reiter Best Paper Award**)

Recursiveness in action

Definition

1. E is conflict-free, and
2. any attacker y is counter-attacked or itself not acceptable

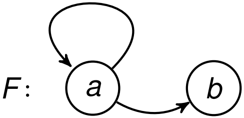


Is $E = \{b\}$ weakly admissible in F?

Recursiveness in action

Definition

1. E is conflict-free, and ✔
2. any attacker y is counter-attacked or itself not acceptable.



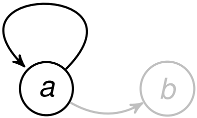
Is $E = \{b\}$ weakly admissible in F?

Recursiveness in action

Definition

1. E is conflict-free, and ✔
2. any attacker y is counter-attacked or itself not acceptable.

F^E:

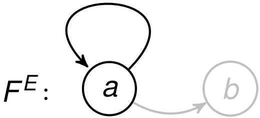


Yes, if a is not contained in a weakly admissible set of F^E.

Recursiveness in action

Definition

1. E is conflict-free, and ✔
2. any attacker y is counter-attacked or itself not acceptable. ✔



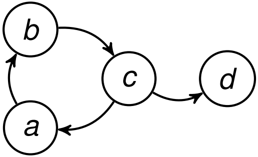
Yes, if a is not contained in a weakly admissible set of F^E.

Recursiveness in action

Definition

1. E is conflict-free, and ✔
2. any attacker y is counter-attacked or itself not acceptable.

$F:$

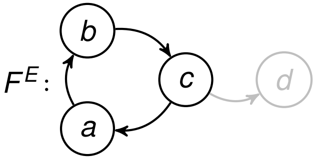


Is $E = \{d\}$ weakly admissible in F?

Recursiveness in action

Definition

1. E is conflict-free, and ✔
2. any attacker y is counter-attacked or itself not acceptable. ✍



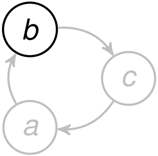
Yes, if c is not contained in a w-admissible set of F^E.

Recursiveness in action

Definition

1. E is conflict-free, and
2. any attacker y is counter-attacked or itself not acceptable.

$(F^E)^{\{c\}}$:

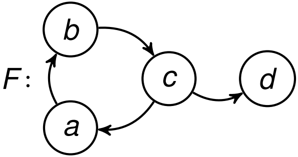


Yes, if b is contained in a w-admissible set of $(F^E)^{\{c\}}$.

Recursiveness in action

Definition

1. E is conflict-free, and ✔
2. any attacker y is counter-attacked or itself not acceptable. ✔



Yes, $E = \{d\}$ is weakly admissible in F.

A Bunch of Semantics

Stable, Semi-stable, Preferred, Complete, Admissible, Grounded, Ideal, Eager, Stage, Cf-zwei, Stage-zwei, Prudent, Naive, Stagle, Strong Admissible, Weak Admissible, Weak Preferred, Weak Complete, Weak Grounded and Conflict-tolerant Semantics

Beyond Reconstruction

Beyond Reconstruction

Argumentation, a phenomenon we are all familiar with, arises in response to, or in anticipation of, a real or imagined difference of opinion.

[van Eemeren and Verheij, 2017]

Beyond Reconstruction

Argumentation, a phenomenon we are all familiar with, arises in response to, or in anticipation of, a real or imagined difference of opinion.

[van Eemeren and Verheij, 2017]

dialogues, persuasion, negotiation, decision making ...

Beyond Reconstruction

Argumentation, a phenomenon we are all familiar with, arises in response to, or in anticipation of, a real or imagined difference of opinion.

[van Eemeren and Verheij, 2017]

dialogues, persuasion, negotiation, decision making ...

Computational argumentation deals with formal models of an argument as well as approaches and techniques formalizing inference on the basis of arguments.

Limitations of Dung AFs

They cannot express:

- support between arguments
- collective attacks
- attacks on attacks
- values
- preferences
- ...

Limitations of Dung AFs

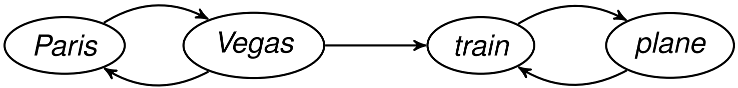
They cannot express:

- support between arguments
- collective attacks
- attacks on attacks
- values
- preferences
- ...

$\Rightarrow$ need for more expressive frameworks

Abstract Dialectical Frameworks

- most powerful generalization of Dung AFs
- use acceptance conditions instead of attack arcs

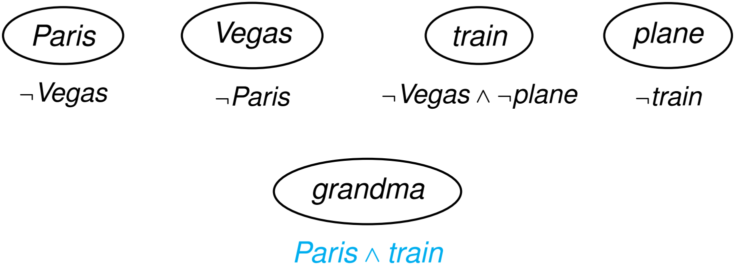


Abstract Dialectical Frameworks

- most powerful generalization of Dung AFs
- use acceptance conditions instead of attack arcs

Abstract Dialectical Frameworks

- most powerful generalization of Dung AFs
- use acceptance conditions instead of attack arcs



"Grandma lives in a suburb of Paris, which would be a stop on the train route."

Abstract Dialectical Frameworks

- semantics rely on the $\mathcal{C}_D$-operator

Definition

For an ADF $D = (S, P)$ we define $\mathcal{C}_D : \mathcal{V}_3^D \mapsto \mathcal{V}_3^D$ as

$$\mathcal{C}_D(v) : S \mapsto \{t, f, u\} \text{ with } s \mapsto \sqcap_i \left\{ w(\phi_s) \mid w \in [v]_2^D \right\}.$$

Abstract Dialectical Frameworks

- semantics rely on the $\mathcal{C}_D$-operator

Definition

For an ADF $D = (S, P)$ we define $\mathcal{C}_D : \mathcal{V}_3^D \mapsto \mathcal{V}_3^D$ as

$$\mathcal{C}_D(v) : S \mapsto \{t, f, u\} \text{ with } s \mapsto \sqcap_i \left\{ w(\phi_s) \mid w \in [v]_2^D \right\}.$$

- $\mathcal{V}_3^D = \{v \mid v : S \to \{t, f, u\}\}$ (three-valued interpretation)
- the information order $<_i$ is defined as: $u <_i t$ and $u <_i f$
- $\leq_i$ is the reflexive closure and $\sqcap_i$ is the consensus, i.e.

$$t \sqcap_i t = t, \quad f \sqcap_i f = f, \quad \text{and } u \text{ otherwise}$$

- $[v]_2^D = \{w \mid w : S \to \{t, f\}, v \leq_i w\}$ (two-valued completions)

Abstract Dialectical Frameworks

- semantics rely on the $\mathcal{C}_D$-operator

Definition

For an ADF $D = (S, P)$ we define $\mathcal{C}_D : \mathcal{V}_3^D \mapsto \mathcal{V}_3^D$ as

$$\mathcal{C}_D(v) : S \mapsto \{t, f, u\} \text{ with } s \mapsto \sqcap_i \left\{ w(\phi_s) \mid w \in [v]_2^D \right\}.$$

Definition

Given an ADF $D = (S, P)$ and $v \in \mathcal{V}_3^D$.

1. $v \in ad(D)$ iff $v \leq_i \mathcal{C}_D(v)$,
2. $v \in co(D)$ iff $v = \mathcal{C}_D(v)$,
3. $v \in pr(D)$ iff v is $\leq_i$-maximal in $co(D)$, and
4. $v \in gr(D)$ iff v is $\leq_i$-least in $co(D)$.

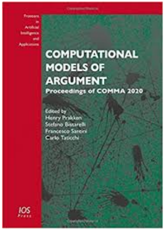
Vorlesung “Formale Argumentation” - Planned Topics

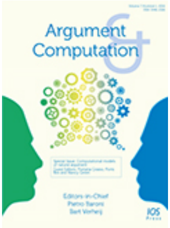
1. Semantics and Properties
2. Complexity
3. Weak Admissibility
4. Realizability and Maximal Numbers
5. Replaceability
6. Intertranslatability
7. Modularity and Splitting
8. Enforcement, Repair and Forgetting
9. Labelling-based Semantics and ADFs
10. Structured Argumentation
11. ABA and others

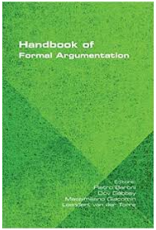
Argumentation (is a vibrant research area) in AI

keyword at major AI conferences

dedicated conferences, journals, handbooks and competitions







Vorlesung “Formale Argumentation”

1. Einführung und Überblick

Ringo Baumann
Professur für Formale Argumentation
und Logisches Schließen

04. April 2024
Leipzig