Instantiating Knowledge Bases in Abstract Dialectical Frameworks

Hannes Strass

Computer Science Institute
Leipzig University, Germany

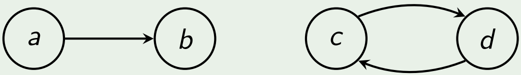
CLIMA XIV
16 September 2013

Motivation: AFs

State of the art in abstract argumentation

Abstract Argumentation Frameworks (AFs)

- syntactically: directed graphs



- conceptually: nodes are arguments, edges denote attacks between arguments
- semantics: determine which arguments can be accepted together
- used as target language for translations from more expressive languages (e.g. ASPIC)
- drawback: can only express attack

Motivation: ADFs

Recent improvements

Abstract Dialectical Frameworks (ADFs)

- generalise AFs, arguments are now called *statements*
- can also (although less directly) be visualised as graphs
- edges express that there is some relationship between the two statements
- relationship need not be "attack", precise nature specified by *acceptance condition* for each statement
- acceptance condition specifies status of node given status of direct predecessors

Outline

1. Background
   - Defeasible Theory Bases
   - Abstract Argumentation Frameworks
   - Abstract Dialectical Frameworks
2. From DTBs to AFs
   - General Scheme
   - Caminada & Amgoud: ASPIC
   - Wyner, Bench-Capon & Dunne
3. From DTBs to ADFs
4. Conclusion

Outline

Defeasible Theories

consist of strict and defeasible rules

- Lit ... set of literals $p, q, \neg q$
- semantical negation $\bar{\cdot}$ with $\overline{p} = \neg p$ and $\overline{\neg p} = p$
- $S \subseteq Lit$ is consistent iff there is no $\psi \in Lit$ with $\psi, \neg\psi \in S$
- strict rule: $r : \phi_1, \ldots, \phi_n \rightarrow \psi$
- defeasible rule: $r : \phi_1, \ldots, \phi_n \Rightarrow \psi$
- ψ ... rule head, $\phi_1, \ldots, \phi_n$ ... rule body, r ... rule name
- defeasible theory base (DTB): $(Lit, StrInf, DefInf)$
  - $StrInf$ ... set of strict rules
  - $DefInf$ ... set of defeasible rules

a/ka defeasible theory, a/ka theory base

Abstract Argumentation Frameworks[1]

are for determining acceptance of abstract arguments

Definition: Abstract Argumentation Framework

- pair $F = (A, R)$
- A . . . set of arguments
- $R \subseteq A \times A$ . . . attack relation

Abstract Argumentation Semantics

- labelling (valuation) of the arguments as accepted (true), rejected (false) or undecided (unknown)
- e.g. stable labelling: no attacks between accepted arguments, every rejected argument is attacked by some accepted one

[1] Phan Minh Dung. "On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games". In: *Artificial Intelligence* 77 (2 1995), pages 321–358.

Abstract Dialectical Frameworks[2]

Syntax

Definition: Abstract Dialectical Framework

An abstract dialectical framework (ADF) is a triple $D = (S, L, C)$,

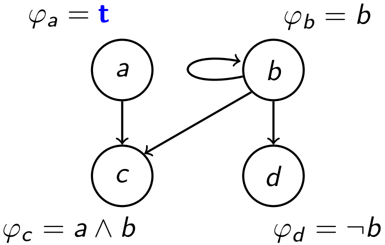
- S ... set of statements (correspond to AF arguments)
- $L \subseteq S \times S$ ... links $\qquad (par(s) = L^{-1}(s))$
- $C = \{C_s\}_{s \in S}$ ... acceptance conditions

- links denote some kind of dependency relation
- acceptance condition: Boolean function $C_s : 2^{par(s)} \to \{\mathbf{t}, \mathbf{f}\}$
- here: C_s often specified by propositional formula φ_s

[2] Gerhard Brewka and Stefan Woltran. "Abstract Dialectical Frameworks". In: *Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (KR)*. 2010, pages 102–111.

Abstract Dialectical Frameworks

Example



Abstract Dialectical Frameworks

Semantics

Truth values, interpretations

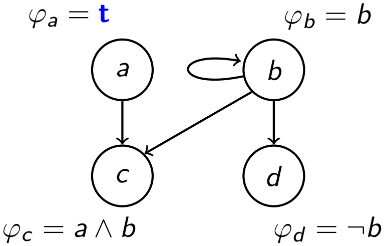
- truth values: true $\mathbf{t}$, false $\mathbf{f}$, unknown $\mathbf{u}$
- interpretation: $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$
- interpretations can be represented as consistent sets of literals

Semantics

- two-valued v is a *model* of D iff $v(s) = v(\varphi_s)$ for all $s \in S$
- there is also a *stable model semantics*, which checks for support cycles

Abstract Dialectical Frameworks

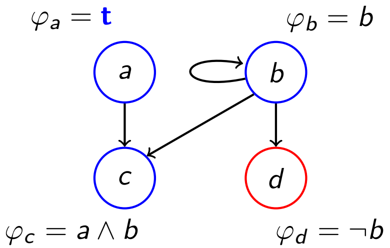
Semantics: Example



- models:
  - $v_1 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}, d \mapsto \mathbf{f}\}$
  - $v_2 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{f}, d \mapsto \mathbf{t}\}$

Abstract Dialectical Frameworks

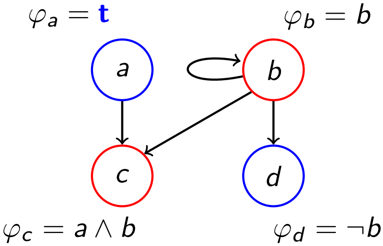
Semantics: Example



- models:
  - $v_1 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}, d \mapsto \mathbf{f}\}$
  - $v_2 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{f}, d \mapsto \mathbf{t}\}$

Abstract Dialectical Frameworks

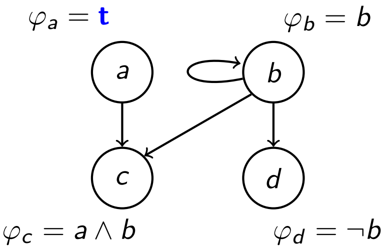
Semantics: Example



- models:
  - $v_1 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}, d \mapsto \mathbf{f}\}$
  - $v_2 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{f}, d \mapsto \mathbf{t}\}$

Abstract Dialectical Frameworks

Semantics: Example



- models:
  - $v_1 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{t}, c \mapsto \mathbf{t}, d \mapsto \mathbf{f}\}$ (not stable)
  - $v_2 = \{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{f}, d \mapsto \mathbf{t}\}$ (stable)

Outline

1. Background
   - Defeasible Theory Bases
   - Abstract Argumentation Frameworks
   - Abstract Dialectical Frameworks
2. From DTBs to AFs
   - General Scheme
   - Caminada & Amgoud: ASPIC
   - Wyner, Bench-Capon & Dunne
3. From DTBs to ADFs
4. Conclusion

From DTBs to AFs, General Scheme

how it works

1. construct arguments
2. construct attacks
3. determine accepted arguments of AF
4. determine accepted conclusions of original DTB

From DTBs to AFs, ASPIC-style[3]

structured arguments

- arguments are constructed inductively from rules
- base case: rule "$\Rrightarrow \psi$" with empty body leads to argument $A = [\Rrightarrow \psi]$ with conclusion ψ
- induction: arguments $A_1, \ldots, A_n$ with conclusions $\phi_1, \ldots, \phi_n$ and rule $r : \phi_1, \ldots, \phi_n \Rrightarrow \psi$ lead to argument $A = [A_1, \ldots, A_n \Rrightarrow \psi]$ with conclusion ψ (A_i are subarguments of A)
- argument is *strict* if only strict rules used for construction (otherwise the argument is defeasible)

[3] Martin Caminada and Leila Amgoud. "On the evaluation of argumentation formalisms". In: *Artificial Intelligence* 171.5–6 (2007), pages 286–310.

From DTBs to AFs, ASPIC-style

rebuts, undercuts

- two possible reasons for attacks between arguments
- rebut: A rebuts B if subargument A' of A has conclusion ψ and *defeasible* subargument B' of B has conclusion $\overline{\psi}$
- undercut: A undercuts B if B uses defeasible rule r and subargument A' of A disputes applicability of r
- will only look at rebut here

From DTBs to AFs, ASPIC-style

Example

- w . . . John wears something that looks like a wedding ring
- g . . . John often goes out late with his friends
- m . . . John is married
- b . . . John is a bachelor
- h . . . John has a spouse
- $StrInf = \{r_1 :\to w, \quad r_2 :\to g, \quad r_3 : b \to \neg h, \quad r_4 : m \to h\}$
- $DefInf = \{r_5 : w \Rightarrow m, \quad r_6 : g \Rightarrow b\}$
- ASPIC: $S = \{w, g, m, b\}$ are sceptical conclusions ("John is a married bachelor"), indirectly inconsistent

Rationality Postulates

Intend to capture semantically “rational” behaviour

- given a DTB and its argumentation translation:

Direct Consistency

Any model of the translation is consistent.

Closure

Any model is closed under strict rules.

Indirect Consistency

Any model’s closure under strict rules is consistent.

Direct translation[4]

from DTBs to AFs

- "C&A conflate different senses of the term *argument*"
- "subarguments and defeat in terms of subarguments are problematic departures from Dung [1995]"

direct translation: literals and rule names become arguments

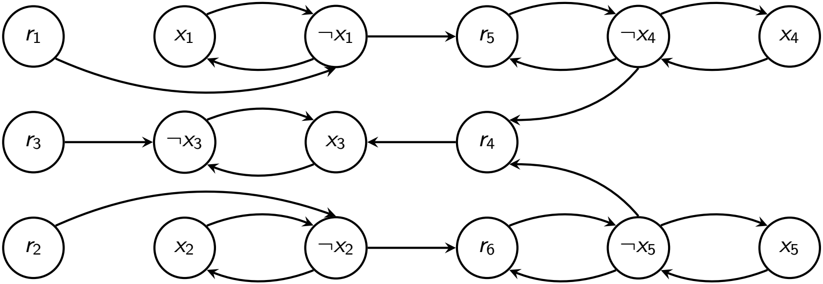
- opposite literals attack each other
- rules are attacked by the negations of their body literals
- defeasible rules are attacked by the negation of their head
- all rules attack the negation of their head

[4] Adam Wyner, Trevor Bench-Capon, and Paul Dunne. "Instantiating knowledge bases in abstract argumentation frameworks". In: *Proceedings of the AAAI Fall Symposium – The Uses of Computational Argumentation*. 2009.

Translation of Wyner et al.

Example with an undesired stable labelling

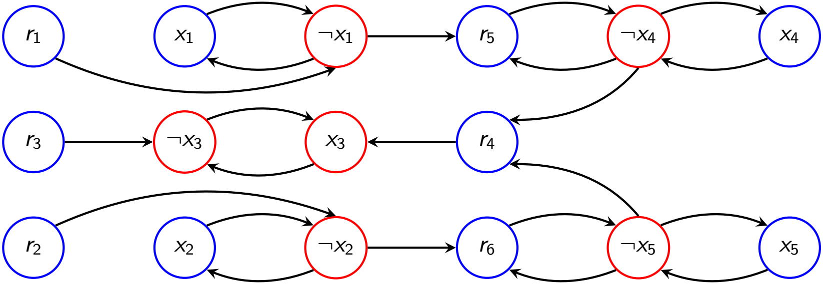
$$Lit = \{x_1, x_2, x_3, x_4, x_5, \neg x_1, \neg x_2, \neg x_3, \neg x_4, \neg x_5\}$$
$$StrInf = \{r_1 :\to x_1, \quad r_2 :\to x_2, \quad r_3 :\to x_3, \quad r_4 : x_4, x_5 \to \neg x_3\}$$
$$DefInf = \{r_5 : x_1 \Rightarrow x_4, \quad r_6 : x_2 \Rightarrow x_5\}$$



Translation of Wyner et al.

Example with an undesired stable labelling

$$Lit = \{x_1, x_2, x_3, x_4, x_5, \neg x_1, \neg x_2, \neg x_3, \neg x_4, \neg x_5\}$$
$$StrInf = \{r_1 :\to x_1, \quad r_2 :\to x_2, \quad r_3 :\to x_3, \quad r_4 : x_4, x_5 \to \neg x_3\}$$
$$DefInf = \{r_5 : x_1 \Rightarrow x_4, \quad r_6 : x_2 \Rightarrow x_5\}$$



Outline

From DTBs to ADFs

statements

- statements: literals, rule names, "negated" rule names
  $S = \mathit{Lit} \cup \{r, \text{-}r \mid r : \phi_1, \ldots, \phi_n \Rrightarrow \psi \in \mathit{StrInf} \cup \mathit{DefInf}\}$
- for $\psi \in \mathit{Lit}$,

$$\varphi_\psi = \neg[\overline{\psi}] \wedge \bigvee_{r:\phi_1,\ldots,\phi_n \Rrightarrow \psi \in \mathit{StrInf} \cup \mathit{DefInf}} [r]$$

- for a strict rule $r : \phi_1, \ldots, \phi_n \to \psi \in \mathit{StrInf}$,

$$\varphi_r = [\phi_1] \wedge \ldots \wedge [\phi_n], \qquad \varphi_{\text{-}r} = [\phi_1] \wedge \ldots \wedge [\phi_n] \wedge \neg[\psi] \wedge \neg[\text{-}r]$$

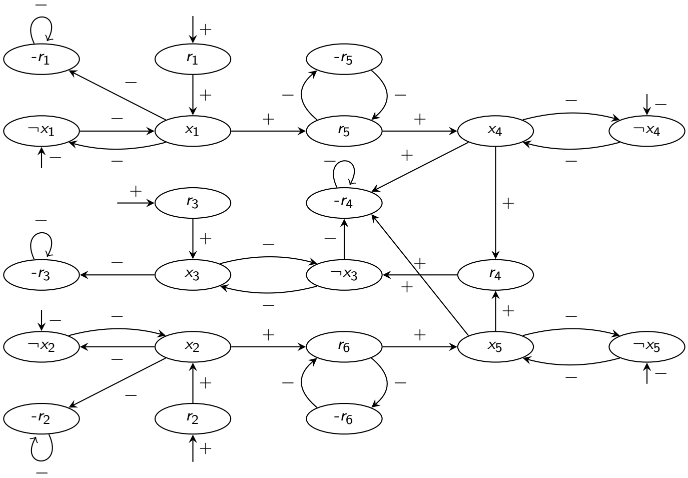
- for a defeasible rule $r : \phi_1, \ldots, \phi_n \Rightarrow \psi \in \mathit{DefInf}$, we define

$$\varphi_r = [\phi_1] \wedge \ldots \wedge [\phi_n] \wedge \neg[\overline{\psi}] \wedge \neg[\text{-}r] \quad \text{and} \quad \varphi_{\text{-}r} = \neg[r]$$

From DTBs to ADFs: Previous Example

$$StrInf = \{r_1 :\rightarrow x_1, \quad r_2 :\rightarrow x_2, \quad r_3 :\rightarrow x_3, \quad r_4 : x_4, x_5 \rightarrow \neg x_3\}$$

$$DefInf = \{r_5 : x_1 \Rightarrow x_4, \quad r_6 : x_2 \Rightarrow x_5\}$$



Some properties of the translation

- support cycles through rules can be detected:
  $\mathit{DefInf} = \{r_1 : \mathit{rain} \Rightarrow \mathit{wet}, r_2 : \mathit{wet} \Rightarrow \mathit{rain}\}$
- postulates are fulfilled: direct/indirect consistency, closure
- can be computed in polynomial time, blowup in size is quadratic, blowup in number of arguments is linear

Outline

Conclusion

of the talk

- reviewed translations from DTBs to AFs
- presented translation from DTBs to ADFs
- future work:
  - allow rules that use rule names as atoms
  - try to avoid integrity constraints, make use of three-valued semantics

Thank you!