TECHNISCHE UNIVERSITÄT DRESDEN

Artificial Intelligence, Computational Logic

DECOMPOSING ABSTRACT DIALECTICAL FRAMEWORKS

Sarah Gaggl and Hannes Strass

Pitlochry, 12th September 2014

DRESDEN concept Exzellenz aus Wissenschaft und Kultur

Motivation

- Computational complexity of semantics for ADFs is in general higher than for AFs [Strass and Wallner, 2014].
- Algorithms based on SCC-recursive schema for AF semantics show significant performance gain [Cerutti et.al. KR 2014].
- We propose a similar approach based on a recursive decomposition along SCCs.
- Allows to define *cf2* and *stage2* semantics for ADFs.

Motivation

- Computational complexity of semantics for ADFs is in general higher than for AFs [Strass and Wallner, 2014].
- Algorithms based on SCC-recursive schema for AF semantics show significant performance gain [Cerutti et.al. KR 2014].
- We propose a similar approach based on a recursive decomposition along SCCs.
- Allows to define *cf2* and *stage2* semantics for ADFs.

Main Difference to AFs

1. Acceptance conditions of statements in sub-frameworks may still depend on statements not contained in sub-framework.
2. Elimination of redundancies from links and acceptance formulas.
3. Propagation of truth values to subsequent SCCs.

Agenda

ADFs - The Formal Framework

- Like AFs, use graph to describe dependencies among nodes.
- Unlike AFs, allow individual acceptance condition for each node.
- Assigns $\mathbf{t}$(rue) or $\mathbf{f}$(alse) depending on status of parents.

Definition

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

- S is a set of statements (positions, nodes),
- $L \subseteq S \times S$ is a set of links,
- $C = \{C_s\}_{s \in S}$ is a set of total functions $C_s : 2^{par(s)} \to \{\mathbf{t}, \mathbf{f}\}$, one for each statement s. C_s is called acceptance condition of s.

Semantics

Definition

Let φ be a propositional formula over vocabulary S and for an $M \subseteq S$ let $v : M \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ be a three-valued interpretation.
The partial valuation of φ by v is $\varphi^v = \varphi[p/\mathbf{t} : v(p) = \mathbf{t}][p/\mathbf{f} : v(p) = \mathbf{f}]$.

Definition

Let $D = (S, L, C)$ be an ADF. A three-valued interpretation v is

- conflict-free iff for all $s \in S$ we have:
  - $v(s) = \mathbf{t}$ implies that φ_s^v is satisfiable,
  - $v(s) = \mathbf{f}$ implies that φ_s^v is unsatisfiable;
- naive iff it is $\leq_i$-maximal with respect to being conflict-free;

Where $\leq_i$ is a partial order over the truth values (resp. interpretations), i.e. $\mathbf{u} <_i \mathbf{t}$ and $\mathbf{u} <_i \mathbf{f}$.

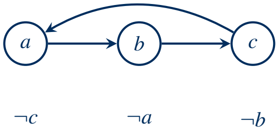
Semantics ctd.

Definition

Let $D = (S, L, C)$ be an ADF. The partial valuation of φ by v is
$\varphi^v = \varphi[p/\mathbf{t} : v(p) = \mathbf{t}][p/\mathbf{f} : v(p) = \mathbf{f}]$.
A three-valued interpretation v is

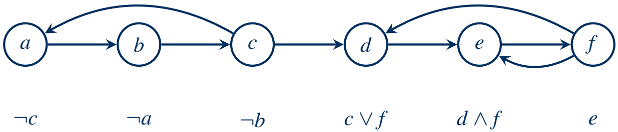
- conflict-free iff for all $s \in S$ we have:
  - $v(s) = \mathbf{t}$ implies that φ_s^v is satisfiable,
  - $v(s) = \mathbf{f}$ implies that φ_s^v is unsatisfiable;
- naive iff it is $\leq_i$-maximal with respect to being conflict-free;

Example



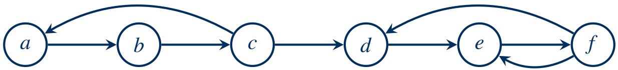
$v = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{u}, c \mapsto \mathbf{t}\}$ is conflict-free, as $\varphi_a^v = \neg\mathbf{t}$ is unsatisfiable and $\varphi_c^v = \neg b$ is satisfiable.

Sub-Frameworks and Redundancies



- independent set $ind_D(\emptyset) = \{a, b, c\} = M_0$
- independent modulo M_0: $ind_D(M_0) = \{a, b, c, d, e, f\} = S$
- M independent set: sub-framework $D|_M = (M, L \cap (M \times M), \{\varphi_s\}_{s \in M})$

Sub-Frameworks and Redundancies



$\neg c$ $\neg a$ $\neg b$ $c \vee f$ $d \wedge f$ e

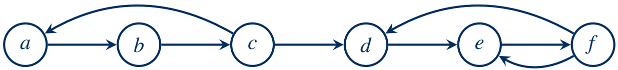
- independent set $ind_D(\emptyset) = \{a, b, c\} = M_0$
- independent modulo M_0: $ind_D(M_0) = \{a, b, c, d, e, f\} = S$
- M independent set: sub-framework $D|_M = (M, L \cap (M \times M), \{\varphi_s\}_{s \in M})$

- Redundancies can change dependencies between statements.
- If (r, s) is redundant then r has no influence on the truth value of φ_s whatsoever.

Example

Consider $\varphi_s = a \vee (b \wedge c)$ and the interpretation $v = \{a \mapsto \mathbf{u}, b \mapsto \mathbf{f}, c \mapsto \mathbf{u}\}$.
$\varphi_s^v = a \vee (\mathbf{f} \wedge c)$

Sub-Frameworks and Redundancies



$\neg c$ $\neg a$ $\neg b$ $c \vee f$ $d \wedge f$ e

- independent set $ind_D(\emptyset) = \{a, b, c\} = M_0$
- independent modulo M_0: $ind_D(M_0) = \{a, b, c, d, e, f\} = S$
- M independent set: sub-framework $D|_M = (M, L \cap (M \times M), \{\varphi_s\}_{s \in M})$

- Redundancies can change dependencies between statements.
- If (r, s) is redundant then r has no influence on the truth value of φ_s whatsoever.

Example

Consider $\varphi_s = a \vee (b \wedge c)$ and the interpretation $v = \{a \mapsto \mathbf{u}, b \mapsto \mathbf{f}, c \mapsto \mathbf{u}\}$.
$\varphi_s^v = a \vee (\mathbf{f} \wedge c) \equiv a$ c has no influence

Reduced ADF

Given an ADF $D = (S, L, C)$, an independent set $M \subseteq S$ and an interpretation $v : M \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$. The ADF D reduced with v on M is obtained by:

- adapt the acceptance condition of statement s to
  - $\mathbf{t}$ (resp. $\mathbf{f}$) if $v(s) = \mathbf{t}$ (resp. $v(s) = \mathbf{f}$)
  - $\neg s$ if $v(s) = \mathbf{u}$
  - partial valuation φ_s^v for remaining statements and if r is redundant in φ_s^v, replace r with $\mathbf{t}$
- remove redundant links
- add links $\{(s, s) \mid v(s) = \mathbf{u}\}$

Reduced ADF

Given an ADF $D = (S, L, C)$, an independent set $M \subseteq S$ and an interpretation $v : M \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$. The ADF D reduced with v on M is obtained by:

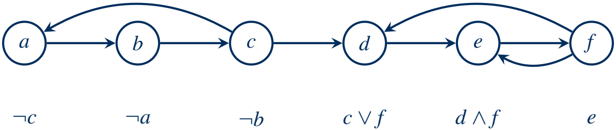
- adapt the acceptance condition of statement s to
  - $\mathbf{t}$ (resp. $\mathbf{f}$) if $v(s) = \mathbf{t}$ (resp. $v(s) = \mathbf{f}$)
  - $\neg s$ if $v(s) = \mathbf{u}$
  - partial valuation φ_s^v for remaining statements and if r is redundant in φ_s^v, replace r with $\mathbf{t}$
- remove redundant links
- add links $\{(s, s) \mid v(s) = \mathbf{u}\}$

Procedure

For a semantics σ and an ADF D, we obtain the σ_2 interpretations recursively by applying $\sigma_2(D) = \sigma_2(ind_D(\emptyset), D)$ by:

1. Start with all statements independent modulo $\emptyset$, i.e. $M_0 = ind_D(\emptyset)$
2. Compute all σ-interpretations of sub-framework $D|_{M_0}$
3. For each σ-interpretation w of $D|_{M_0}$ compute the reduced ADF
4. Call Step 1 with reduced ADF and $M_1 = ind_D(M_0)$.

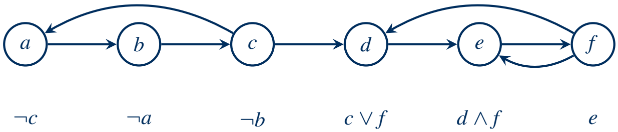
Example



$nai_2(D) = nai_2(ind_D(\emptyset), D)$

1. $ind_D(\emptyset) = \{a, b, c\} = M_0$

Example

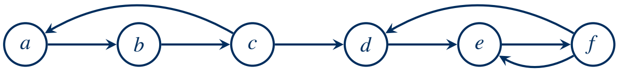


$nai_2(D) = nai_2(ind_D(\emptyset), D)$

1. $ind_D(\emptyset) = \{a, b, c\} = M_0$
2. Then we obtain $nai(D|_{M_0}) = \{v_0, v_1, v_2\}$:

$$
\begin{aligned}
v_0 &= \{a \mapsto \mathbf{u}, b \mapsto \mathbf{t}, c \mapsto \mathbf{f}\}, \\
v_1 &= \{a \mapsto \mathbf{f}, b \mapsto \mathbf{u}, c \mapsto \mathbf{t}\}, \\
v_2 &= \{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{u}\}.
\end{aligned}
$$

Example

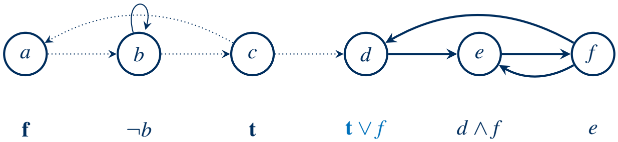


$\neg c$ $\neg a$ $\neg b$ $c \vee f$ $d \wedge f$ e

$nai_2(D) = nai_2(ind_D(\emptyset), D)$

1. $ind_D(\emptyset) = \{a, b, c\} = M_0$
2. $v_1 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{u}, c \mapsto \mathbf{t}\}$

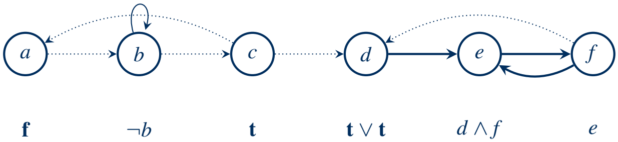
Example



$nai_2(D) = nai_2(ind_D(\emptyset), D)$

1. $ind_D(\emptyset) = \{a, b, c\} = M_0$
2. $v_1 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{u}, c \mapsto \mathbf{t}\}$
3. Reduced ADF with $\mathbf{t} \vee f \equiv \mathbf{t}$, thus link (f, d) is redundant

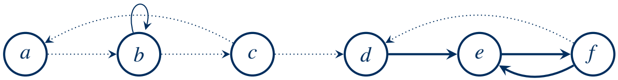
Example



$nai_2(D) = nai_2(ind_D(\emptyset), D)$

1. $ind_D(\emptyset) = \{a, b, c\} = M_0$
2. $v_1 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{u}, c \mapsto \mathbf{t}\}$
3. Reduced ADF

Example



$\mathbf{f}$ $\quad \neg b \quad$ $\mathbf{t}$ $\quad \mathbf{t} \vee \mathbf{t} \quad$ $d \wedge f$ $\quad e$

$nai_2(D) = nai_2(ind_D(\emptyset), D)$

1. $ind_D(\emptyset) = \{a, b, c\} = M_0$
2. $v_1 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{u}, c \mapsto \mathbf{t}\}$
3. Reduced ADF
4. Call Step 1 with reduced ADF D_1 and $M_1 = ind_{D_1}(M_0)$.

Example

$\mathbf{f}$ $\quad \neg b$ $\quad \mathbf{t}$ $\quad \mathbf{t} \vee \mathbf{t}$ $\quad d \wedge f$ $\quad e$

$nai_2(M_1, D_1)$

1. $M_1 = ind_{D_1}(M_0) = \{a, b, c, d\}$

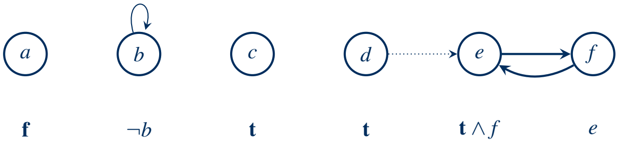
Example

$\mathbf{f}$ $\neg b$ $\mathbf{t}$ $\mathbf{t} \vee \mathbf{t}$ $d \wedge f$ e

$nai_2(M_1, D_1)$

1. $M_1 = ind_{D_1}(M_0) = \{a, b, c, d\}$
2. $nai(D|_{M_1}) = v_3 = v_1 \cup \{d \mapsto \mathbf{t}\}$

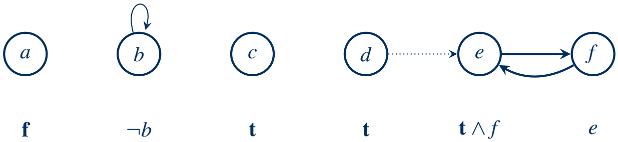
Example



$nai_2(M_1, D_1)$

1. $M_1 = ind_{D_1}(M_0) = \{a, b, c, d\}$
2. $nai(D|_{M_1}) = v_3 = v_1 \cup \{d \mapsto \mathbf{t}\}$
3. Reduced ADF

Example



$nai_2(M_1, D_1)$

1. $M_1 = ind_{D_1}(M_0) = \{a, b, c, d\}$
2. $nai(D|_{M_1}) = v_3 = v_1 \cup \{d \mapsto \mathbf{t}\}$
3. Reduced ADF
4. Call Step 1 with reduced ADF D_2 and $M_2 = ind_{D_2}(M_1)$

Example

a	b	c	d	e	f
$\mathbf{f}$	$\neg b$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t} \wedge f$	e

$nai_2(M_2, D_2)$

1. $M_2 = ind_{D_2}(M_1) = \{a, b, c, d, e, f\} = S$

Example

$\mathbf{f}$ $\quad \neg b$ $\quad \mathbf{t}$ $\quad \mathbf{t}$ $\quad \mathbf{t} \wedge f$ $\quad e$

$nai_2(M_2, D_2)$

1. $M_2 = ind_{D_2}(M_1) = \{a, b, c, d, e, f\} = S$
2. $nai(D_2) = \{v_4, v_6\}$:

$$v_4 = v_3 \cup \{e \mapsto \mathbf{t}, f \mapsto \mathbf{t}\}, \qquad v_6 = v_3 \cup \{e \mapsto \mathbf{f}, f \mapsto \mathbf{f}\}.$$

Main Theorem

Theorem

1. Let $\sigma \in \{\mathit{cfi}, \mathit{adm}, \mathit{pre}, \mathit{com}, \mathit{mod}\}$. Then $\sigma \leq \sigma_2$.
2. Let $\sigma \in \{\mathit{nai}, \mathit{stg}\}$. Then $\sigma \not\leq \sigma_2$.
3. Let $\sigma \in \{\mathit{cfi}, \mathit{nai}, \mathit{adm}, \mathit{pre}, \mathit{com}, \mathit{mod}\}$. Then $\sigma_2 \leq \sigma$.
4. Let $\sigma \in \{\mathit{stg}\}$. Then $\sigma_2 \not\leq \sigma$.

Conclusion

- We proposed a decomposition schema for semantics for ADFs.
- We introduced nai_2 and stg_2 for ADFs.
- Due to the relation of ADFs to logic programms we also get nai_2 and stg_2 semantics for LPs.
- Also in the paper: composing ADFs.

Future Work

- Analysis of the complexity of the approach.
- Implementation.
- Study splittings of ADFs and equivalences.

References

Leila Amgoud and Claudette Cayrol and Marie-Christine Lagasquie and Pierre Livet,
On Bipolarity in Argumentation Frameworks
International Journal of Intelligent Systems 23(10): 1062–1093 (2008)

P. Baroni, F. Cerutti, M. Giacomin and G. Guida.
AFRA: Argumentation Framework with Recursive Attacks.
Int. J. Approx. Reasoning 52(1): 19–37 (2011).

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

G. Brewka and S. Woltran.
Abstract Dialectical Frameworks.
KR 2010, pp. 102–111, AAAI Press, 2010.

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):321–358.

S. Modgil.
Reasoning about Preferences in Argumentation Frameworks.
Artif. Intell. 173(9-10): 910–934 (2009).

H. Strass and J. Wallner.
Analyzing the Computational Complexity of Abstract Dialectical Frameworks via Approximation Fixpoint Theory.
KR 2014, AAAI Press, 2014.