Université Paris Cité

Knowledge Representation and Reasoning

Argumentation Dynamics

Jean-Guy Mailly

M1 Computer Science

Outline

Motivation

- "Natural" argumentation is inherently dynamic: in a debate, new arguments and attacks are added step by step
- Two kinds of approaches:
  - Strategic: knowing the current state of the debate, and some target (set of) argument(s), can I do some actions that guarantee that my target becomes accepted?
  - Computational: knowing the current state of the debate, and the next action (*e.g.* addition of argument and attacks), can I efficiently compute the new extensions without re-computing everything?

Extension Enforcement

Defined by [Baumann and Brewka 2010]

Strict Enforcement

$$\left.\begin{array}{c} F = \langle A, R \rangle \\ E \subseteq A \end{array}\right\} \quad \Longrightarrow \quad F' = \langle A', R' \rangle$$

such that E is an extension of F' for a given semantics

Non-Strict Enforcement

$$\left.\begin{array}{c} F = \langle A, R \rangle \\ E \subseteq A \end{array}\right\} \quad \Longrightarrow \quad F' = \langle A', R' \rangle$$

such that E is included in an extension of F' for a given semantics

- There may be constraints on how to choose F'

Université Paris Cité

Expansions [Baumann and Brewka 2010]

Definition

Given $F = \langle A, R \rangle, F' = \langle A', R' \rangle$, F' is a **normal expansion** of F iff $A \subset A'$ and $R' \cap (A \times A) = R$

Expansions [Baumann and Brewka 2010]

Definition

Given $F = \langle A, R \rangle, F' = \langle A', R' \rangle$, F' is a **normal expansion** of F iff $A \subset A'$ and $R' \cap (A \times A) = R$

- Intuitively, a normal expansion is a new AF which adds new arguments and attacks, but does not change the attacks between former arguments

Expansions [Baumann and Brewka 2010]

Definition

Given $F = \langle A, R \rangle, F' = \langle A', R' \rangle$, F' is a **normal expansion** of F iff $A \subset A'$ and $R' \cap (A \times A) = R$

- Intuitively, a normal expansion is a new AF which adds new arguments and attacks, but does not change the attacks between former arguments
- Specific cases of normal expansion:

Expansions [Baumann and Brewka 2010]

Definition

Given $F = \langle A, R \rangle$, $F' = \langle A', R' \rangle$, F' is a **normal expansion** of F iff $A \subset A'$ and $R' \cap (A \times A) = R$

- Intuitively, a normal expansion is a new AF which adds new arguments and attacks, but does not change the attacks between former arguments
- Specific cases of normal expansion:

weak expansion
: adds only weak arguments, *i.e.* arguments which don't attack the former arguments

Expansions [Baumann and Brewka 2010]

Definition

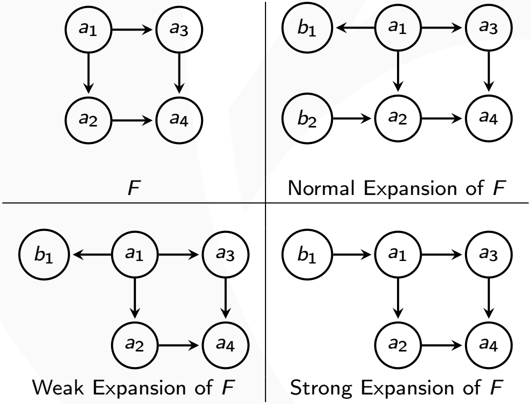
Given $F = \langle A, R \rangle, F' = \langle A', R' \rangle$, F' is a **normal expansion** of F iff $A \subset A'$ and $R' \cap (A \times A) = R$

- Intuitively, a normal expansion is a new AF which adds new arguments and attacks, but does not change the attacks between former arguments
- Specific cases of normal expansion:

weak expansion: adds only weak arguments, *i.e.* arguments which don't attack the former arguments

strong expansion: adds only strong arguments, *i.e.* arguments which are not attacked by the former arguments

Example: Normal, Weak, Strong Expansions



Enforcement Based on Expansions

Defined by [Baumann and Brewka 2010]

Strict Normal (resp. Weak, Strong) Enforcement

$$\left.\begin{array}{c} F = \langle A, R\rangle \\ E \subseteq A \end{array}\right\} \implies F' = \langle A', R'\rangle \text{ such that}$$

- E is an extension of F'
- F' is a normal (resp. weak, strong) expansion of F

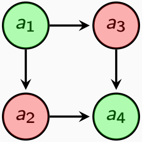
Non-Strict Normal (resp. Weak, Strong) Enforcement

$$\left.\begin{array}{c} F = \langle A, R\rangle \\ E \subseteq A \end{array}\right\} \implies F' = \langle A', R'\rangle \text{ such that}$$

- E is included in an extension of F'
- F' is a normal (resp. weak, strong) expansion of F

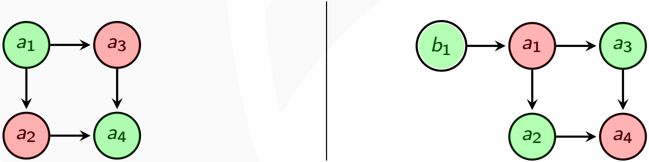
Example of Strong Enforcement

- Using $\sigma = st$, how to enforce $E = \{a_2, a_3\}$ in F?



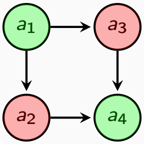
Example of Strong Enforcement

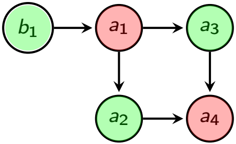
- Using $\sigma = st$, how to enforce $E = \{a_2, a_3\}$ in F?



Example of Strong Enforcement

- Using $\sigma = st$, how to enforce $E = \{a_2, a_3\}$ in F?





- Non-strict enforcement is always possible with strong expansion, but it may not be the case for strict enforcement

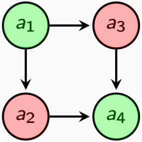
Argument-Fixed and General Enforcement

Defined in [Coste-Marquis *et al* 2015]

- **Argument-fixed enforcement**: perform a strict or non-strict enforcement without modifying the set of arguments (modifying attacks is possible)
- **General enforcement**: perform a strict or non-strict enforcement by any possible means (adding arguments, modifying attacks)

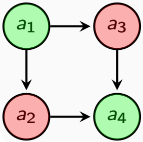
Example: Argument-Fixed Enforcement

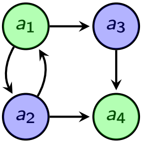
- Using $\sigma = st$, how to enforce $E = \{a_2, a_3\}$ in F?



Example: Argument-Fixed Enforcement

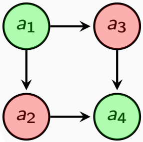
- Using $\sigma = st$, how to enforce $E = \{a_2, a_3\}$ in F?

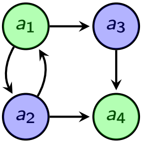




Example: Argument-Fixed Enforcement

- Using $\sigma = st$, how to enforce $E = \{a_2, a_3\}$ in F?





- Strict enforcement is always possible with argument-fixed/general enforcement

Minimal Change [Baumman 2012]

- **Minimal enforcement**: F' must be as close as possible from F, closeness is measured with Hamming distance

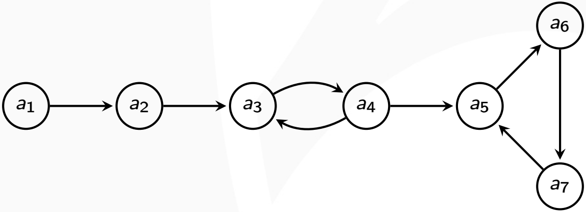
$$d_H(F, F') = |(R \setminus R') \cup (R' \setminus R)|$$

Semantic Change [Doutre and Mailly 2017]

Idea:

- For performing the enforcement, we have the choice between several semantics
- Choose the semantics that allows to enforce the extension with minimal change of the graph

Example



- Current semantics: $\sigma = st$, $st(F) = \{\{a_1, a_4, a_6\}\}$
- Goal: enforcing $E = \{a_1, a_3\}$
- Without semantic change: the graph has to be modified
- With semantic change: switch semantics from *st* to *pr*, since $E \in pr(F) = \{\{a_1, a_3\}, \{a_1, a_4, a_6\}\}$. **No change of the graph at all**

Computation

- Efficient approaches for computing the result of (minimal) enforcement, based on optimization problems related to SAT
  - pseudo-Boolean constraints [Coste-Marquis *et al* 2015]
  - MaxSAT [Wallner *et al* 2017]

References

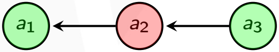
- R. Baumann and G. Brewka, *Expanding Argumentation Frameworks: Enforcing and Monotonicity Results*. COMMA'10, pp. 75-86, 2010.
- S. Coste-Marquis, S. Konieczny, J.-G. Mailly and P. Marquis, *Extension Enforcement in Abstract Argumentation as an Optimization Problem*. IJCAI'15, pp 2876-2882, 2015.
- R. Baumann, *What Does it Take to Enforce an Argument? Minimal Change in abstract Argumentation*. ECAI'12, pp 127-132, 2012.
- S. Doutre and J.-G. Mailly, *Semantic Change and Extension Enforcement in Abstract Argumentation*. SUM'17, 2017.
- J. P. Wallner, A. Niskanen and M. Järvisalo, *Complexity Results and Algorithms for Extension Enforcement in Abstract Argumentation*. J. Artif. Intell. Res. 60: 1-40, 2017.

Dynamic Computation: Intuition

- When the AF is updated, detect which part of it is impacted by the update
- Re-compute only the extensions for this part, and combine it with the "old" extension of the rest
- Example: the extension of this AF is $\{a_1, a_3\}$

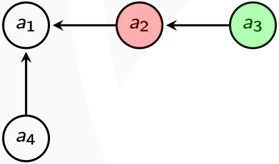
Dynamic Computation: Intuition

- When the AF is updated, detect which part of it is impacted by the update
- Re-compute only the extensions for this part, and combine it with the "old" extension of the rest
- Example: the extension of this AF is $\{a_1, a_3\}$



Dynamic Computation: Intuition

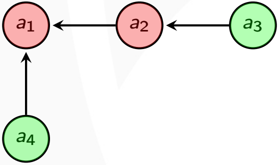
- When the AF is updated, detect which part of it is impacted by the update
- Re-compute only the extensions for this part, and combine it with the "old" extension of the rest
- Example: the extension of this AF is $\{a_1, a_3\}$



- Update: new argument a_4 attacks a_1
- Arguments a_2 and a_3 are not impacted by the new arguments: compute only the status of arguments for a_1 and a_4, and combine it with the fact that a_3 is accepted, and a_2 rejected

Dynamic Computation: Intuition

- When the AF is updated, detect which part of it is impacted by the update
- Re-compute only the extensions for this part, and combine it with the "old" extension of the rest
- Example: the extension of this AF is $\{a_1, a_3\}$



- Update: new argument a_4 attacks a_1
- Arguments a_2 and a_3 are not impacted by the new arguments: compute only the status of arguments for a_1 and a_4, and combine it with the fact that a_3 is accepted, and a_2 rejected