Bipolar Argumentation for Supporting Decisions in Software Design

Gianluca Amato Fabio Fioravanti Maria Chiara Meo Francesca Scozzari

Università "G. d'Annunzio" Chieti–Pescara, Italy

SAC 2026 (KRR)

Motivation

- Software design often requires choosing languages, libraries, DBMSs, and configurations.
- Decisions must balance conflicting goals such as security, energy efficiency, and open source compliance.
- The same choice can help one goal and harm another.
- We propose to adopt Bipolar Argumentation Frameworks (BAF) to model these trade-offs in an explainable way.

Outline

Outline

Argumentation

Definition (Argumentation)

Communicative activity of producing and exchanging reasons in order to support claims or defend/challenge positions, especially in situations of doubt or disagreement.

The International Encyclopedia of Communication Theory and Philosophy 2016

Argumentation framework

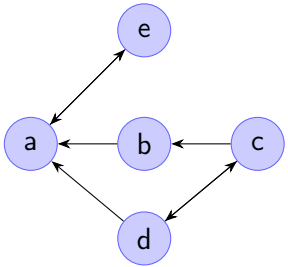
Definition (Argumentation framework — Dung 1995)

An argumentation framework is a directed graph (Arg, Att) where

- Arg is a set of nodes (arguments);
- $\text{Att} \subseteq \text{Arg} \times \text{Arg}$ is the set of edges (attacks).

Nodes are abstraction of (concrete) arguments:

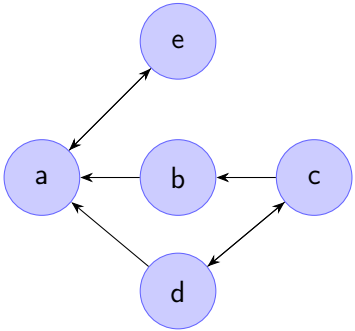
- a : John killed the victim
- b : There is no evidence that John killed the victim
- c : The doorman has seen John at the crime scene
- d : John was teaching a class with many students
- e : John has no motive to kill the victim



Extension based argumentation semantics

Given $AF = (\text{Arg}, \text{Att})$, $S \subseteq \text{Arg}$ is acceptable when:

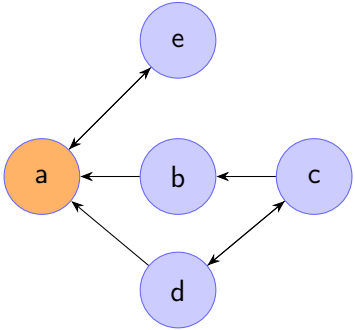
- *Conflict-free semantics*: no two arguments in S attack each other.



Extension based argumentation semantics

Given $AF = (\text{Arg}, \text{Att})$, $S \subseteq \text{Arg}$ is acceptable when:

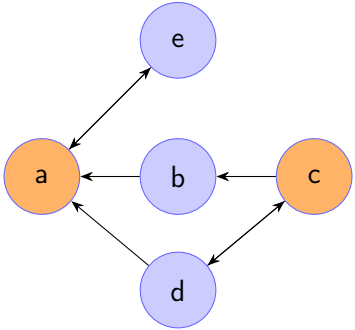
- *Conflict-free semantics*: no two arguments in S attack each other.



Extension based argumentation semantics

Given $AF = (\text{Arg}, \text{Att})$, $S \subseteq \text{Arg}$ is acceptable when:

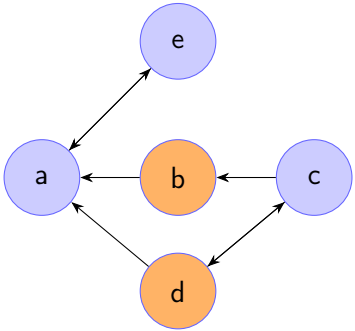
- *Conflict-free semantics*: no two arguments in S attack each other.



Extension based argumentation semantics

Given $AF = (\text{Arg}, \text{Att})$, $S \subseteq \text{Arg}$ is acceptable when:

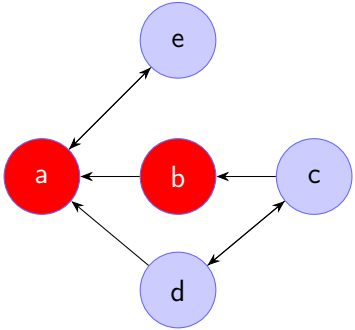
- *Conflict-free semantics*: no two arguments in S attack each other.



Extension based argumentation semantics

Given $AF = (\text{Arg}, \text{Att})$, $S \subseteq \text{Arg}$ is acceptable when:

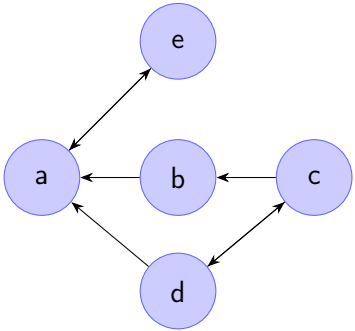
- *Conflict-free semantics*: no two arguments in S attack each other.



Extension based argumentation semantics

Given $AF = (\text{Arg}, \text{Att})$, $S \subseteq \text{Arg}$ is acceptable when:

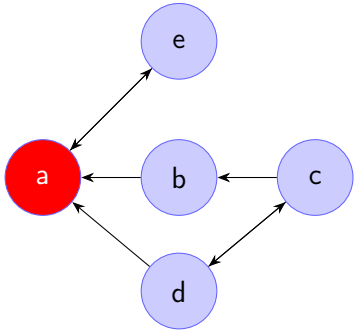
- *Conflict-free semantics*: no two arguments in S attack each other.
- *Admissible semantics*: S is conflict free and defends all its elements.



Extension based argumentation semantics

Given $AF = (\text{Arg}, \text{Att})$, $S \subseteq \text{Arg}$ is acceptable when:

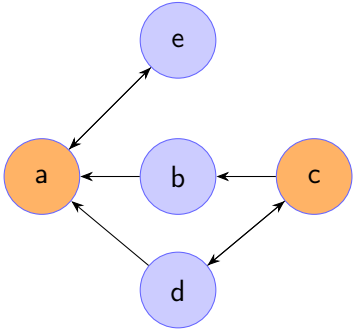
- *Conflict-free semantics*: no two arguments in S attack each other.
- *Admissible semantics*: S is conflict free and defends all its elements.



Extension based argumentation semantics

Given $AF = (\text{Arg}, \text{Att})$, $S \subseteq \text{Arg}$ is acceptable when:

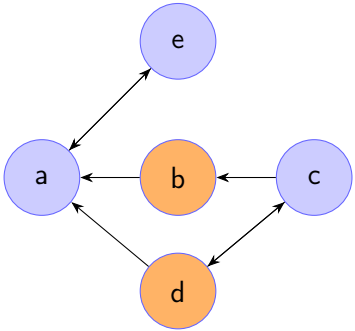
- *Conflict-free semantics*: no two arguments in S attack each other.
- *Admissible semantics*: S is conflict free and defends all its elements.



Extension based argumentation semantics

Given $AF = (\text{Arg}, \text{Att})$, $S \subseteq \text{Arg}$ is acceptable when:

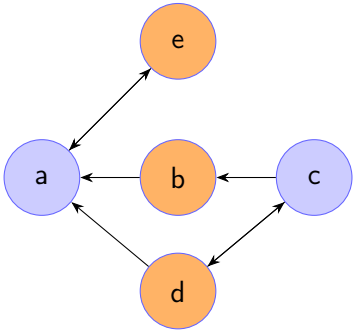
- *Conflict-free semantics*: no two arguments in S attack each other.
- *Admissible semantics*: S is conflict free and defends all its elements.



Extension based argumentation semantics

Given $AF = (\text{Arg}, \text{Att})$, $S \subseteq \text{Arg}$ is acceptable when:

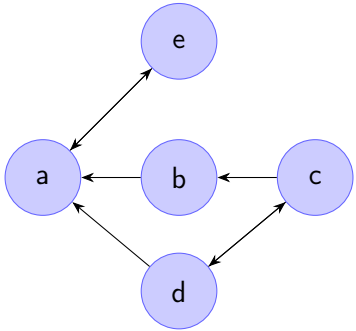
- *Conflict-free semantics*: no two arguments in S attack each other.
- *Admissible semantics*: S is conflict free and defends all its elements.



Extension based argumentation semantics

Given $AF = (\text{Arg}, \text{Att})$, $S \subseteq \text{Arg}$ is acceptable when:

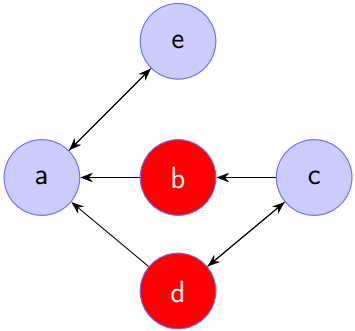
- *Conflict-free semantics*: no two arguments in S attack each other.
- *Admissible semantics*: S is conflict free and defends all its elements.
- *Preferred semantics*: S is a maximal admissible set of arguments.



Extension based argumentation semantics

Given $AF = (\text{Arg}, \text{Att})$, $S \subseteq \text{Arg}$ is acceptable when:

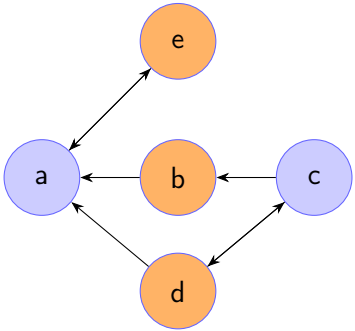
- *Conflict-free semantics*: no two arguments in S attack each other.
- *Admissible semantics*: S is conflict free and defends all its elements.
- *Preferred semantics*: S is a maximal admissible set of arguments.



Extension based argumentation semantics

Given $AF = (\text{Arg}, \text{Att})$, $S \subseteq \text{Arg}$ is acceptable when:

- *Conflict-free semantics*: no two arguments in S attack each other.
- *Admissible semantics*: S is conflict free and defends all its elements.
- *Preferred semantics*: S is a maximal admissible set of arguments.

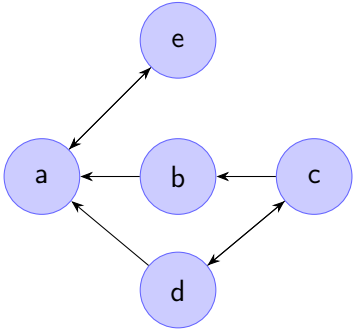


Extension based argumentation semantics

Given $AF = (\text{Arg}, \text{Att})$, $S \subseteq \text{Arg}$ is acceptable when:

- *Conflict-free semantics*: no two arguments in S attack each other.
- *Admissible semantics*: S is conflict free and defends all its elements.
- *Preferred semantics*: S is a maximal admissible set of arguments.

Acceptable sets a are generally called extensions.



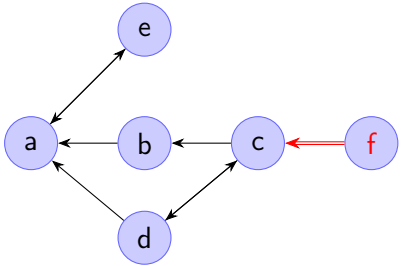
Introducing support between arguments

Definition (Bipolar argumentation framework — Cayrol & Lagasquie-Schiex 2005)

A bipolar argumentation framework is a tuple $BAF = (\text{Arg}, \text{Att}, \text{Supp})$ where

- (Arg, Att) is an argumentation framework;
- $\text{Supp} \subseteq \text{Arg} \times \text{Arg}$ is the support relation.

a : John killed the victim
b : There is no evidence that John killed the victim
c : The doorman has seen John at the crime scene
d : John was teaching a class with many students
e : John has no motive to kill the victim
f : The doorman has excellent eyesight



Outline

Framework for technology selection

We model technology selection as a BAF:

- application and configuration relations, as well as goals, are mapped to arguments;
- positive and negative dependencies among them are captured through support and attack relations, respectively.

This should allow us to:

- modeling technical incompatibilities and preferences;
- reasoning under conflicting goals;
- justifying and explaining the final design decision.

Arguments

- `use(Product, Context)` for design choices.
  - `use(mysql, db)`
  - `use(pgsql, db)`
  - `use(php, pl)`
- `conf(Product, Context, Option, Value)` for configuration choices.
  - `conf(mysql, db, tls, yes)`
- `goal(Goal)` for project objectives.
  - `goal(green)`
  - `goal(opensource)`

Attacks

- **Mutually exclusive technologies**
  - `use(php,pl)` $\rightarrow$ `use(python,pl)`
  - `use(mysql,db)` $\rightarrow$ `use(pgsql,db)`
- **Mutually exclusive configurations**
  - `conf(mysql,db,tls,yes)` $\rightarrow$ `conf(mysql,db,tls,no)`
- **Incompatibility between technologies**
  - `use(mysqli,dblib)` $\rightarrow$ `use(pgsql,db)`
- **Goal-based attacks**
  - `use(oracle,db)` $\rightarrow$ `goal(opensource)`
  - `use(python,pl)` $\rightarrow$ `goal(green)`

Supports

- **Goal-based support**
  - `use(mysql,db)` $\Rightarrow$ `goal(opensource)`.
  - `conf(_,dblib,tls,yes)` $\Rightarrow$ `goal(security)`
- **Requirement-based support**
  - `use(django,webframework)` $\Rightarrow$ `use(python,pl)`
- **Goal-level support**
  - `goal(security)` $\Rightarrow$ `goal(privacy)`

Reasoning Task

- Compute the extensions of the BAF
  - Extensions represent coherent design choices.
  - Possibly focus on extensions containing a target goal, such as `goal(security)`.
- Implementation
  - Aspartix (`https://www.dbai.tuwien.ac.at/proj/argumentation/systempage/`);
  - a collection of ASP programs for either Clingo or DLV;
  - the bipolar framework is only supported with DLV.

Results of experiment

Two examples using c-preferred extensions (like preferred extensions but closed under support):

Extension 1: Green-oriented

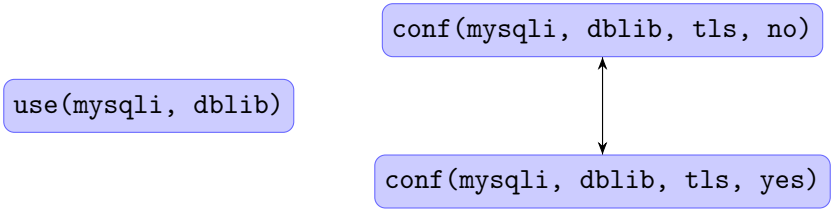
```
goal(opensource), goal(green), use(php,pl),
use(mysql,db), use(mysqli,dblib),
conf(mysqli,dblib,tls,no)
```

Extension 2: Security-oriented

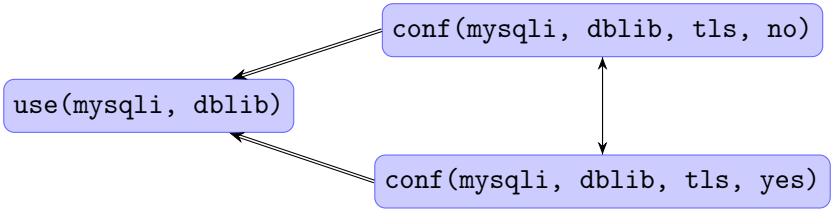
```
goal(opensource), goal(security), goal(privacy),
use(php,pl), use(mysql,db), use(mysqli,dblib),
conf(mysqli,dblib,tls,yes)
```

In both cases, open source is preserved; the main trade-off is **green vs security/privacy**.

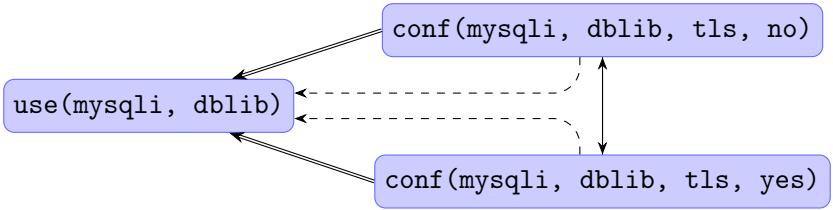
Support semantics is too strong



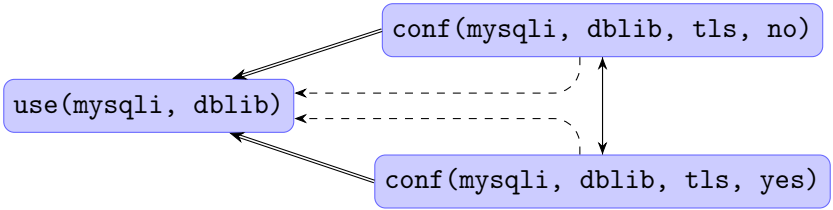
Support semantics is too strong



Support semantics is too strong



Support semantics is too strong



`use(mysqli, dblib)` does not appear in any admissible or preferred extension!

Future Work

- find a less stringent semantics for the support relation;
- scalability evaluation on larger case studies;
- argumentation frameworks with preferences;
- multi-criteria optimization over goals.

Thank you