

Multi-Agent Systems

Argumentation

Albert-Ludwigs-Universität Freiburg



UNI
FREIBURG

Bernhard Nebel, Rolf Bergdoll, and Thorsten Engesser
Winter Term 2019/20



- Modeling agents exchanging arguments
 - Argumentation frameworks
 - Semantics
 - Algorithms



- A: My government cannot negotiate with your government because your government does not even recognize my government.
- B: Your government does not recognize my government either.
- A: But your government is a terrorist government.
- Which arguments should be accepted?



- **A:** Ralph goes fishing, because it is Sunday.
- **B:** Ralph does not go fishing, because it is Mother's day, so he visits his parents.
- **C:** Ralph cannot visit his parents, because it is a leap year, so they are on vacation.

- Which arguments should be accepted?



- A statement is accepted if it can be successfully defended against attacking arguments.

Definition (Argument)

An **argument** is a pair (S, φ) , such that S is a set of formulae and φ can be derived from S . S is also called the **support** for the **claim** φ .

Definition (Attack)

Two definitions of attack:

Undercut Argument $A_1 = (S_1, \varphi_1)$ **undercuts** argument $A_2 = (S_2, \varphi_2)$ iff $\neg\varphi_2$ can be derived from S_1 .

Rebuttal Argument $A_1 = (S_1, \varphi_1)$ **rebutts** argument $A_2 = (S_2, \varphi_2)$ iff $\varphi_1 \equiv \neg\varphi_2$.

We can decide what to believe while looking at arguments at the abstract level (Dung, 1995):

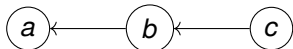
- Disregarding internal structures of arguments
- Focus on the attack relation between arguments
(a, b, c, d, \dots): a **attacks** b or $a \rightsquigarrow b$
- Not concerned with the origin of arguments or the attack relation

Abstract argumentation framework

An **argumentation framework** is a pair $\mathcal{AF} = (Arg, \rightsquigarrow)$ where Arg is a set of arguments and $\rightsquigarrow \subseteq Arg \times Arg$. We say that $a \in Arg$ attacks $b \in Arg$ iff $(a, b) \in \rightsquigarrow$.

- Remember:
 - **A**: Ralph goes fishing, because it is Sunday.
 - **B**: Ralph does not go fishing, because it is Mother's day, so he visits his parents.
 - **C**: Ralph cannot visit his parents, because it is a leap year, so they are on vacation.
- Representation as an argumentation framework:

$$\mathcal{AF} = \langle \{a, b, c\}, \{(b, a), (c, b)\} \rangle,$$



Definition: Labelling

Let $\mathcal{AF} = (Arg, \rightsquigarrow)$ be an argumentation framework. A **labelling** of \mathcal{AF} is a total function $Lab : Arg \rightarrow \{in, out, undec\}$. The set of all labellings will be denoted by $\mathcal{L}(\mathcal{AF})$.

- $in(Lab) = \{a \mid Lab(a) = \mathbf{in}\}$
- $out(Lab) = \{a \mid Lab(a) = \mathbf{out}\}$
- $undec(Lab) = \{a \mid Lab(a) = \mathbf{undec}\}$

- To refer to a labelling Lab we will also write $\langle in(Lab), out(Lab), undec(Lab) \rangle$

$$\mathcal{AF} = \langle \{a, b, c\}, \{(b, a), (c, b)\} \rangle,$$



$$\mathcal{L}(\mathcal{AF}) = \{ \langle \emptyset, \emptyset, \{a, b, c\} \rangle, \langle \emptyset, \{a\}, \{b, c\} \rangle \dots \}$$

- How to identify the appropriate labellings?
- E.g., we do not want to accept both a and b , thus if $Lab(a) = \mathbf{in}$ then $Lab(b) \neq \mathbf{in}$.

Definition: Admissible labelling



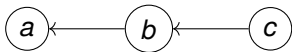
Definition

Let Lab be a labelling of argumentation framework AF . An **in**-labelled argument is said to be **legally in** iff all its attackers are labelled **out**. An **out**-labelled argument is said to be **legally out** iff it has at least one attacker that is labelled **in**.

Definition

Let AF be an argumentation framework. An **admissible labelling** is a labelling where each **in**-labelled argument is legally **in** and each **out**-labelled argument is legally **out**.

$$\mathcal{AF} = \langle \{a, b, c\}, \{(b, a), (c, b)\} \rangle,$$



Admissible labellings

- $\langle \emptyset, \emptyset, \{a, b, c\} \rangle$
- $\langle \{a, c\}, \{b\}, \emptyset \rangle$

Definition

Given an argumentation framework $\mathcal{AF} = (\text{Arg}, \rightsquigarrow)$, a **labelling semantics** S associates with \mathcal{AF} a subset of $\mathcal{L}(\mathcal{AF})$, denoted as $L_S(\mathcal{AF})$.

Definition

Let $\mathcal{AF} = (Arg, \rightsquigarrow)$ be an argumentation framework and $Lab : Arg \rightarrow \{in, out, undec\}$ be a total function. We say that Lab is a **complete labelling** iff it satisfies the following:

$$\forall a \in Arg : (Lab(a) = \mathbf{out} \leftrightarrow \exists b \in Arg : (b \rightsquigarrow a \wedge Lab(b) = \mathbf{in}))$$

$$\forall a \in Arg : (Lab(a) = \mathbf{in} \leftrightarrow \forall b \in Arg : (b \rightsquigarrow a \rightarrow Lab(b) = \mathbf{out}))$$

$$\mathcal{AF} = \langle \{a, b, c\}, \{(b, a), (c, b)\} \rangle,$$



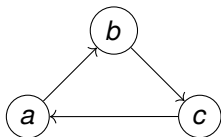
Complete labellings:

1 $\langle \{a, c\}, \{b\}, \emptyset \rangle$

Why not $\langle \emptyset, \emptyset, \{a, b, c\} \rangle$?

- **A:** Bert says that Ernie is unreliable, therefore everything that Ernie says cannot be relied on.
- **B:** Ernie says that Elmo is unreliable, therefore everything that Elmo says cannot be relied on.
- **C:** Elmo says that Bert is unreliable, therefore everything that Bert says cannot be relied on.

$$\mathcal{AF} = \langle \{a, b, c\}, \{(a, b), (b, c), (c, a)\} \rangle,$$



Complete labellings:

1 $\mathcal{Lab}_1 : \langle \emptyset, \emptyset, \{a, b, c\} \rangle$

- **A**: Nixon is a pacifist, because he is a Quaker.
- **B**: Nixon is not a pacifist, because he is a Republican.

$$\mathcal{AF} = \langle \{a, b\}, \{(a, b), (b, a)\} \rangle,$$



Complete labellings:

- 1 $Lab_1 : \langle \emptyset, \emptyset, \{a, b\} \rangle$
- 2 $Lab_2 : \langle \{a\}, \{b\}, \emptyset \rangle$
- 3 $Lab_3 : \langle \{b\}, \{a\}, \emptyset \rangle$

⇒ Three reasonable positions a rational agent can take.



Definition

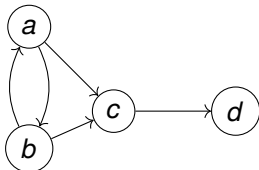
Let \mathcal{AF} be an argumentation framework. The **grounded labelling** of \mathcal{AF} is a complete labelling Lab where $in(Lab)$ is minimal w.r.t. set inclusion.

- Grounded semantics picks the complete labelling with minimal **in**, minimal **out**, and maximal **undec**.
- Intuitively, the arguments in **in** are those that must be accepted by every rational agent.
- These arguments are in the **in** set of every complete labelling.
- The grounded labelling is unique.
- It is the least fixpoint of an operator which assigns in each step **in** to all legally **in**-nodes and **out** to all legally **out**-nodes.

Definition

Let \mathcal{AF} be an argumentation framework. The **preferred labelling** of \mathcal{AF} is a complete labelling Lab where $in(Lab)$ is maximal w.r.t. set inclusion.

- Preferred semantics picks the complete labelling with maximal **in**, maximal **out**, and minimal **undec**.
- For every argumentation framework one or more preferred labellings exists.



- Grounded labelling: $\langle \emptyset, \emptyset, \{a, b, c, d\} \rangle$
- Preferred labellings: $\langle \{a, d\}, \{b, c\}, \emptyset \rangle, \langle b, d \rangle, \{a, c\}, \emptyset$

Observe: Grounded labelling is not among the preferred labellings and none of the preferred labellings is the grounded labelling. Also, it is not the case that the grounded labelling coincides with the intersection of all preferred labellings.

Definition

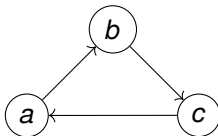
Let Lab be a labelling of an argumentation framework \mathcal{AF} .
 Lab is a **stable labelling** of \mathcal{AF} iff it is a complete labelling with $\mathbf{undec}(Lab) = \emptyset$.

- Stable semantics decides for every argument if it is **in** or **out**, no **undec**.
- As it minimizes **undec** it maximizes **in** and **out**. Thus, every stable labelling is a preferred labelling.
- But not vice versa: Whereas a preferred labelling always exists, the existence of a stable labelling is not guaranteed.



Complete labellings:

$$1 \text{ } Lab_1 : \langle \emptyset, \emptyset, \{a\} \rangle$$



Complete labellings:

$$1 \text{ } Lab_2 : \langle \emptyset, \emptyset, \{a, b, c\} \rangle$$

$\Rightarrow Lab_1, Lab_2$ are complete, grounded, preferred, but not stable.

Restrictions on complete labelling	Resulting semantics
no restrictions	complete semantics
empty undec	stable semantics
maximal in	preferred semantics
maximal out	preferred semantics
maximal undec	grounded semantics
minimal in	grounded semantics
minimal out	grounded semantics

- Every complete labelling is admissible.
- Every grounded labelling is complete.
- Every preferred labelling is complete.
- Every stable labelling is preferred.

- Two central problems:
 - Is an argument **acceptable**?
 - Sceptically acceptable: **in** in all grounded/preferred/stable/... labellings?
 - Credulously acceptable: **in** in at least one grounded/preferred/stable/... labelling?
- Other interesting decision problems:
 - Given some labelling, is it grounded/preferred/stable/... ?
 - How do we generate a grounded/preferred/stable/... labelling?
 - Does there some grounded/preferred/stable/... labelling exist?
 - Does there some nonempty grounded/preferred/stable/... labelling exist?

Focus on two reasoning tasks besides acceptance



- Given an argument A and an argumentation framework \mathcal{AF} , is A in the **in** set of \mathcal{AF} 's grounded labelling?
- Given an argument A and an argumentation framework \mathcal{AF} , is A in the **in** set of some of \mathcal{AF} 's preferred labellings?

Definition

A **partial labelling** is a partial function $Lab : Args \rightarrow \{\mathbf{in}, \mathbf{out}\}$ such that

- if $Lab(A) = \mathbf{in}$ then for each attacker B $Lab(B) = \mathbf{out}$
 - if $Lab(A) = \mathbf{out}$ then for some attacker B $Lab(B) = \mathbf{in}$
-
- Partial labellings are admissible labellings
 - A partial labelling Lab can be extended to a total labelling $Lab' \supseteq Lab$
 - For each total labelling Lab' there exists a partial labelling $Lab \subseteq Lab'$ (just remove the **undec** labels)

Definition

$$\text{extendin}(\mathcal{L}ab) = \mathcal{L}ab \cup \{(A, \mathbf{in}) \mid \forall B [B \rightsquigarrow A \rightarrow \mathcal{L}ab(B) = \mathbf{out}]\}$$
$$\text{extendout}(\mathcal{L}ab) = \mathcal{L}ab \cup \{(A, \mathbf{out}) \mid \exists B [B \rightsquigarrow A \wedge \mathcal{L}ab(B) = \mathbf{in}]\}$$
$$\text{extendinout}(\mathcal{L}ab) = \text{extendin}(\text{extendout}(\mathcal{L}ab))$$

- If $\mathcal{L}ab$ is a partial labelling, then $\text{extendin}(\mathcal{L}ab)$, $\text{extendout}(\mathcal{L}ab)$, $\text{extendinout}(\mathcal{L}ab)$ are partial labellings.

function GROUNDLABELLING(\mathcal{A}, \mathcal{F})

$L \leftarrow \emptyset$

repeat

$L_{old} \leftarrow L$

$L \leftarrow \text{extendout}(L)$

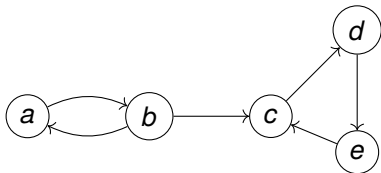
$L \leftarrow \text{extendin}(L)$

until $L = L_{old}$

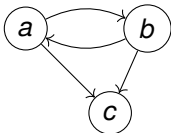
return $L \cup \{(A, \mathbf{undec}) \mid (A, \mathbf{in}) \notin L \text{ and } (A, \mathbf{out}) \notin L\}$

end function

- **Idea:** Take the other's opinion and then derive a contradiction:
 - Proponent (M) makes a statement (A)
 - Opponent (S) derives from A more statements M will be committed to
 - S aims at letting M commit himself to a contradiction
- **Dialog game**
 - M starts and claims the existence of a reasonable position (admissible labelling) in which a particular argument is accepted (labelled **in**).
 - S confronts M with the consequences of M's own position, and asks M to resolve these consequences.
 - S wins if she leads M to a contradiction.
- If M wins then his argument is in the **in** set of an admissible labelling, and thus in the **in** of a preferred labelling.



- M: in(D) *I have an admissible labelling in which D is in*
- S: out(C) *But then in your labelling C is out. Why?*
- M: in(B) *Because B is in*
- S: out(A) *But then A must be out. Why?*
- M: in(B) *Because B is in.*



- M: in(C) *I have an admissible labelling in which C is in*
- S: out(A) *But then in your labelling A is out. Why?*
- M: in(B) *Because B is in*
- S: out(B) *But B must be out!*

Definition

Let $\mathcal{AF} = (Arg, \rightsquigarrow)$ be an argumentation framework. An **admissible discussion** is a sequence of moves

$[\Delta_1, \dots, \Delta_n] (n \geq 0)$ such that:

- each move $\Delta_i (1 \leq i \leq n)$ where i is odd is called M-move and is of the form $in(A)$
- each move $\Delta_i (1 \leq i \leq n)$ where i is even is called S-move and is of the form $out(A)$
- for each S-move $\Delta_i = out(A) (2 \leq i \leq n)$ there exists an M-move $\Delta_j = in(B) (j < i)$ such that A attacks B
- for each M-move $\Delta_i = in(A) (3 \leq i \leq n)$ it holds that Δ_{i-1} is of the form $out(B)$, where A attacks B
- there exist no two S-moves $\Delta_i = \Delta_j$ with $i \neq j$

Definition

An admissible discussion $[\Delta_1, \dots, \Delta_n]$ is said to be **finished** iff

- 1 There exists no Δ_{n+1} such that $[\Delta_1, \dots, \Delta_n, \Delta_{n+1}]$ is an admissible discussion, or there exists a M-move and a S-move containing the same argument
- 2 No subsequence of the discussion is finished.

Definition

A finished admissible discussion is **won** by player S if there exist a M-move and a S-move containing the same argument. Otherwise, it is **won** by the player making the last move.

Theorem [2]

Let g be an admissible discussion won by M and let

$Lab : Ar \rightarrow \{\mathbf{in}, \mathbf{out}, \mathbf{undec}\}$ be a function defined as follows.

For every argument $B \in Ar$:

- $Lab(B) = \mathbf{in}$ if B was labeled in during g
- $Lab(B) = \mathbf{out}$ if B was labeled out during g
- $Lab(B) = \mathbf{undec}$ otherwise

Then Lab is an admissible labelling.

- Thus, if there is a winning game for M defending A then A is in the **in** set of some preferred labelling (because there must be a maximum complete one, containing the admissible one).

Theorem

- 1 *The problem to check whether a given labelling is admissible, complete, grounded, or stable can be decided in polynomial time.*
- 2 *The problem to check whether a given labelling is preferred is coNP-complete.*
- 3 *The problem to check whether a given argumentation system has a stable labelling is NP-complete.*

Proof.

(1) is obvious. (2) Membership: For a given labelling, guess another one, check whether it is a super-labelling. If so, non-preferability has been shown. Hardness follows from complexity results in logic programming and graph theory [6]. (3) Membership obvious, hardness follows from a reduction coming later.

Definition (Credulous Acceptance)

Given $\mathcal{AF} = (Arg, \rightsquigarrow)$ and $a \in Arg$: is a labelled **in** in at least one grounded/preferred/stable/. . . labelling?

Theorem

Deciding credulous acceptance is:

- *NP-complete for stable, admissible, complete, and preferred semantics, and*
- *in P for grounded semantics.*

Proof.

Grounded semantics: Fixpoint construction and check!

Definition (Credulous Acceptance)

Given $\mathcal{AF} = (Arg, \rightsquigarrow)$ and $a \in Arg$: is a labelled **in** in at least one grounded/preferred/stable/... labelling?

Theorem

Deciding credulous acceptance is:

- *NP-complete for stable, admissible, complete, and preferred semantics, and*
- *in P for grounded semantics.*

Proof.

Grounded semantics: Fixpoint construction and check!

Membership in NP: Guess labelling and check whether it satisfies the conditions and the argument is labelled **in**.

A generic reduction from SAT



For $\varphi = \bigwedge_{i=1}^m l_{i_1} \vee l_{i_2} \vee l_{i_3}$ over atoms Z , build $F_\varphi = (A_\varphi, R_\varphi)$ with

$$A_\varphi = Z \cup \hat{Z} \cup \{C_1, \dots, C_m\} \cup \{\varphi\}$$

$$R_\varphi = \{(z, \hat{z}), (\hat{z}, z) \mid z \in Z\} \cup \{(C_i, \varphi) \mid i \in \{1, \dots, m\}\} \cup \\ \{(z, C_i) \mid i \in \{1, \dots, m\}, z \in \{l_{i_1}, l_{i_2}, l_{i_3}\}\} \cup \\ \{(\hat{z}, C_i) \mid i \in \{1, \dots, m\}, \hat{z} \in \{l_{i_1}, l_{i_2}, l_{i_3}\}\}$$

A generic reduction from SAT

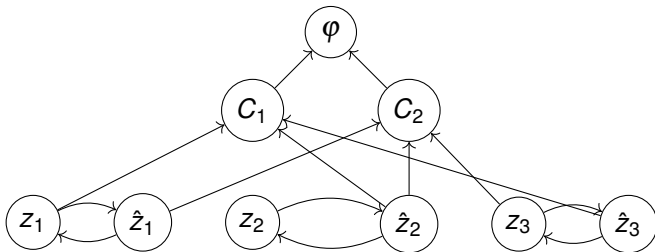


For $\varphi = \bigwedge_{i=1}^m l_{i_1} \vee l_{i_2} \vee l_{i_3}$ over atoms Z , build $F_\varphi = (A_\varphi, R_\varphi)$ with

$$A_\varphi = Z \cup \hat{Z} \cup \{C_1, \dots, C_m\} \cup \{\varphi\}$$

$$R_\varphi = \{(z, \hat{z}), (\hat{z}, z) \mid z \in Z\} \cup \{(C_i, \varphi) \mid i \in \{1, \dots, m\}\} \cup \\ \{(z, C_i) \mid i \in \{1, \dots, m\}, z \in \{l_{i_1}, l_{i_2}, l_{i_3}\}\} \cup \\ \{(\hat{z}, C_i) \mid i \in \{1, \dots, m\}, \hat{z} \in \{l_{i_1}, l_{i_2}, l_{i_3}\}\}$$

Example: Let $\varphi = (z_1 \vee \neg z_2 \vee \neg z_3) \wedge (\neg z_1 \vee \neg z_2 \vee z_3)$



Theorem

The following statements are equivalent:

- 1 φ is satisfiable,
- 2 F_φ has an admissible labelling containing φ as an **in**-node,
- 3 F_φ has a complete labelling containing φ as an **in**-node,
- 4 F_φ has a preferred labelling containing φ as an **in**-node,
- 5 F_φ has a stable labelling containing φ as an **in**-node,

With that, NP-hardness follows.

Definition (Skeptical Acceptance)

Given $\mathcal{AF} = (Arg, \rightsquigarrow)$ and $a \in Arg$: Is a labelled **in** in every grounded/preferred/stable/... labelling?

Theorem

Deciding skeptical acceptance is:

- *co-NP-complete for stable semantics,*
- *computationally trivial for admissible semantics,*
- *in P for complete and grounded semantics, and*
- $\Pi_2^p = \text{co-NP}^{\text{NP}}$ -*complete for preferred semantic.*







Proof.

- 1 Stable semantics: Falsifiability of a DNF formula (which is NP-complete) and the non-membership of an argument is equivalent. So, the complementary problem of membership in every stable labelling must be co-NP-hard. Membership follows from a guess (a labelling) and check non-membership.
- 2 Admissibility semantics: Obvious!
- 3 Complete semantics and grounded semantics: Obvious!
- 4 Solve complementary problem (i.e. non-membership). Guess preferred labelling and check. Note: Deciding whether a labelling is preferrable is not easy: It is already co-NP-complete, i.e., the problem is in NP^{NP} . Hardness proof (reduction from 2-QBF) omitted.

- Can be used to decide what to do next.
- Can be used to find perfect matchings [3]
 - Arg: The couples
 - $(m_1, w_1) \rightsquigarrow (m_2, w_2)$ iff
 - $m_1 = m_2$ and m_1 prefers w_1 to w_2 , or
 - $w_1 = w_2$ and w_1 prefers m_1 to m_2
- Ressource allocation
 - Arg: Pairs $(agent, task)$
 - $(agent_i, task_i) \rightsquigarrow (agent_j, task_j)$ iff one of:
 - $(agent_i, task_i)$ is preferred to $(agent_j, task_j)$
 - $(agent_i, task_i)$ excludes $(agent_j, task_j)$
 - Agent is unable to do $task_j$ (then self attack of $(agent_i, task_i)$)
- Can be used to compute the set of arguments an agent should utter / keep for itself (Persuasion).



- In abstract argumentation systems all arguments are equally strong—relaxation
~> **Preference-based argumentation systems** (e.g., Amgoud et al. 1998f) model preference (weights) of arguments.
- Acceptability of arguments can depend on the target audience (e.g., newspaper vs. scientific article)
~> **Value-based argumentation systems** (Bench-Capon et al, 2003ff)
- Arguments in abstract argumentation systems do not have an internal (logical) structure
~> **Deductive argumentation systems**

-  M. Caminada, A gentle introduction to argumentation semantics. Technical report, University of Luxembourg, Summer 2008.
-  M. Caminada, W. Dvorak, S. Vesic, Preferred semantics as socratic discussion. Journal of Logic and Computation, 2014.
-  P.M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence 77, pp. 321-357, 1995.
-  J.-G. Maily, Dynamics of Argumentation Frameworks, PhD Thesis, 2015.
-  Philippe Besnard and Sylvie Doutre. Checking the acceptability of a set of arguments. In Proceedings of the 10th International Workshop on Non-Monotonic Reasoning (NMR'04), pages 59-64, 2004.
-  Yannis Dimopoulos, Alberto Torres. Graph Theoretical Structures in Logic Programs and Default Theories. Theor. Comput. Sci. 170(1-2): 209-244, 1996.