On computational problems for infinite argumentation frameworks: Hardness of finding acceptable extensions

Luca San Mauro (University of Bari)

AI^3 2024, Free University of Bozen-Bolzano

Joint work with Uri Andrews

Setting the ground

Here's a sobering yet fundamental lesson coming from argumentation theory: *deciding whether to accept an argument is computationally really hard*.

To be more precise, we focus on the admissible, stable, and complete semantics for (Dung-style) argumentation frameworks (AF).

Recall that a conflict-free extension S of a given AF $\mathcal{F} = (A_{\mathcal{F}}, R_{\mathcal{F}})$ is:

- admissible, $S \in ad(\mathcal{F})$, if S is self-defending;
- stable, $S \in stb(\mathcal{F})$, if S attacks all arguments outside itself;
- complete, $S \in co(\mathcal{F})$, if S is admissible and defends no argument outside itself.

The complexity landscape (in the finite setting)

So, here's the complexity of popular decision problems for these semantics:

σ	Cred_σ	Skept_σ	Exist_σ	Nemp_σ	Uni_σ
ad	**NP**-c	trivial	trivial	**NP**-c	**coNP**-c
stb	**NP**-c	**coNP**-c	**NP**-c	**NP**-c	**DP**-c
co	**NP**-c	**P**-c	trivial	**NP**-c	**coNP**-c

$\mathcal{C}$-c denotes completeness for the class $\mathcal{C}$

This brings us to the core question of our research: *What does this table look like when we analyze infinite AFs?*

Formalizing our question

A basic problem that one encounters when attempting to calibrate the algorithmic complexity of infinite AFs is that of describing infinite objects in a finitary way. Fortunately, computability theory offers a wide range of tools designed for this endeavour.

Here, we will concentrate on AFs that are computably presentable, in the sense that there are Turing machines (or, equivalently, modern computer programs) that, in finitely many steps, decide whether a given pair of arguments belongs to the attack relation.

Let's be more formal!

Introducing computable AFs

Let $(\varphi_e)_{e\in\mathbb{N}}$ be a uniform enumeration of all partial computable functions from $\mathbb{N}$ to $\{0,1\}$ and let $\langle\cdot,\cdot\rangle : \mathbb{N}\times\mathbb{N}\to\mathbb{N}$ be a computable bijection.

Definition

- A number e is a computable index for an AF $\mathcal{F} = (A_{\mathcal{F}}, R_{\mathcal{F}})$ with $A_{\mathcal{F}} = \{a_n : n \in \mathbb{N}\}$ if

$$\varphi_e(\langle n,m\rangle) = \begin{cases} 1 & \text{if } a_n \rightarrowtail a_m \\ 0 & \text{otherwise;} \end{cases}$$

- An AF $\mathcal{F}$ is computable, if it has a computable index $e \in \mathbb{N}$.

Formalizing our computational problems

For a semantics σ:

- $\mathrm{Cred}_\sigma^\infty := \{\langle e, n\rangle : (\exists S \in \sigma(\mathcal{F}_e))(a_n \in S)\}$;
- $\mathrm{Skept}_\sigma^\infty := \{\langle e, n\rangle : (\forall S \in \sigma(\mathcal{F}_e))(a_n \in S)\}$;
- $\mathrm{Exist}_\sigma^\infty := \{e : (\exists S \subseteq A_{\mathcal{F}_e}))(S \in \sigma(\mathcal{F}_e))\}$;
- $\mathrm{Nemp}_\sigma^\infty := \{e : (\exists S \in \sigma(\mathcal{F}_e))(S \neq \emptyset)\}$;
- $\mathrm{Uni}_\sigma^\infty := \{e : (\exists ! S \subseteq A_{\mathcal{F}_e})(S \in \sigma(\mathcal{F}_e))\}$.

It turns out that the complexity classes that most naturally match these problems are those of the Σ_1^1 and Π_1^1 sets.

Σ^1_1: The infinitary analog of NP

The Σ^1_1 sets are formally defined as those subsets of $\mathbb{N}$ that are definable in the language of second-order arithmetic using a single second-order existential quantifier ranging over subsets of $\mathbb{N}$. Π^1_1 sets are the complements of Σ^1_1 sets.

Intuition: Just as **NP** allows a search over sets in the finite setting, Σ^1_1 allows a search over sets in the infinite setting. Just as **NP**-complete means that there is no shortcut over searching over all sets, Σ^1_1-complete means that there is no shortcut over searching over all sets.

Complexity in the infinite setting

(Andrews, S.): *The next table collects the promised complexity results about infinite AFs:*

σ	$\mathrm{Cred}_\sigma^\infty$	$\mathrm{Skept}_\sigma^\infty$	$\mathrm{Exist}_\sigma^\infty$	$\mathrm{Nemp}_\sigma^\infty$	$\mathrm{Uni}_\sigma^\infty$
ad	Σ_1^1-c	trivial	trivial	Σ_1^1-c	Π_1^1-c
stb	Σ_1^1-c	Π_1^1-c	Σ_1^1-c	Σ_1^1-c	Π_1^1-c
co	Σ_1^1-c	Π_1^1-c	trivial	Σ_1^1-c	Π_1^1-c

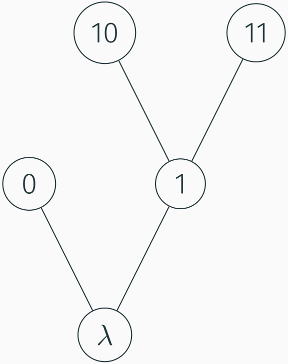
It may be useful to compare it with the one for finite AFs:

σ	Cred_σ	Skept_σ	Exist_σ	Nemp_σ	Uni_σ
ad	**NP**-c	trivial	trivial	**NP**-c	**coNP**-c
stb	**NP**-c	**coNP**-c	**NP**-c	**NP**-c	**DP**-c
co	**NP**-c	**P**-c	trivial	**NP**-c	**coNP**-c

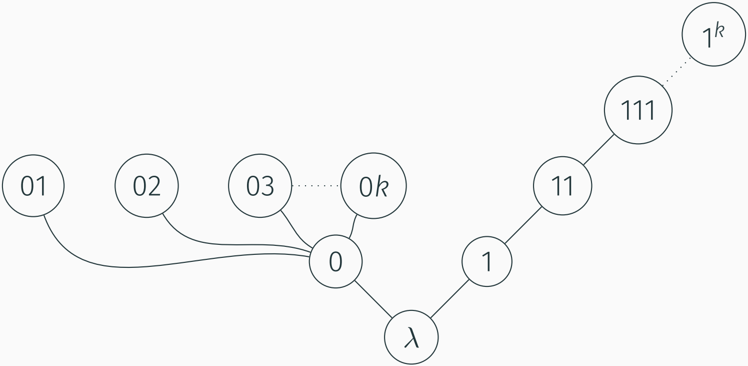
How to show Σ^1_1/Π^1_1-hardness?

Just as you show NP-completeness of a set by reducing a known NP-complete problem (e.g., SAT) to it, we show Σ^1_1-completeness of a set by reducing a known Σ^1_1-complete problem to it. To describe such a problem, we need to introduce trees and paths.

A set $\mathcal{T} \subseteq \mathbb{N}^*$ of finite sequences of natural numbers is a tree if it is closed under prefixes. Recall that λ denotes the empty string. Here are a couple of trees.



The tree $\{\lambda, 0, 1, 10, 11\}$. (Keep this tree in mind!)



The tree $\{\lambda, 0\} \cup \{0k, 1^k : k > 0\}$.

.

Finding paths through trees

A path through $\mathcal{T}$ is an *infinite* sequence of natural numbers all of whose prefixes are in $\mathcal{T}$.

The problem of determining if a tree in $\mathbb{N}^*$ has paths is as hard as it could be:

Theorem (Kleene)

A set $X \subseteq \mathbb{N}$ is Σ^1_1 iff there is a computable sequence of computable trees $(\mathcal{T}^X_n)_{n\in\mathbb{N}}$ so that $n \in X$ iff $\mathcal{T}^X_n$ has a path.

Corollary: *The set of indices for computable trees which have a path is Σ^1_1-complete.*

Coding trees into AFs: Don't read this slide!

Given any tree $\mathcal{T} \subseteq \mathbb{N}^*$, we define an AF $\mathcal{F}^\mathcal{T} = (A^\mathcal{T}, R^\mathcal{T})$.

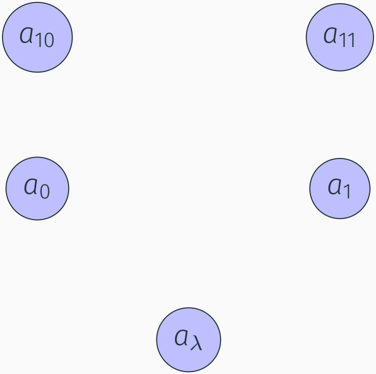
The set of arguments $A^\mathcal{T}$ of $\mathcal{F}^\mathcal{T}$ is computable and consists of $\{a_\sigma : \sigma \in \mathcal{T}\} \cup \{b_\sigma : \sigma \in \mathcal{T}\} \cup \{c\}$. The attack relation $R^\mathcal{T}$ of $\mathcal{F}^\mathcal{T}$ contains all and only the following edges:

For all $\sigma \in \mathcal{T}$,

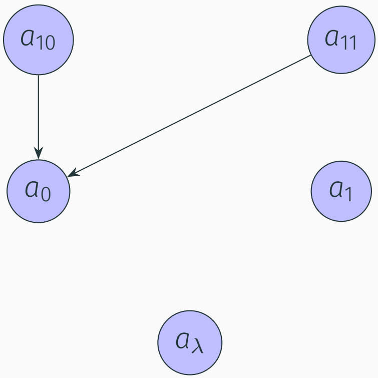
1. $b_\sigma \rightarrowtail b_\sigma$;
2. $b_\sigma \rightarrowtail a_\sigma$;
3. $a_\sigma \rightarrowtail b_\tau$, if $|\sigma| = |\tau| + 1$;
4. $a_\sigma \rightarrowtail a_\tau$, if $|\sigma| = |\tau| + 1$ and $\tau \not\preceq \sigma$;
5. $c \rightarrowtail a_\tau$ for every $\tau \in \mathcal{T}$;
6. $a_\lambda \rightarrowtail c$.

How about a picture-example?

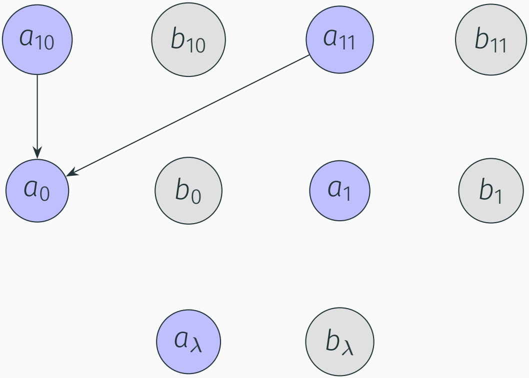
Enconding the tree $\{\lambda, 0, 1, 10, 11\}$ into an AF:



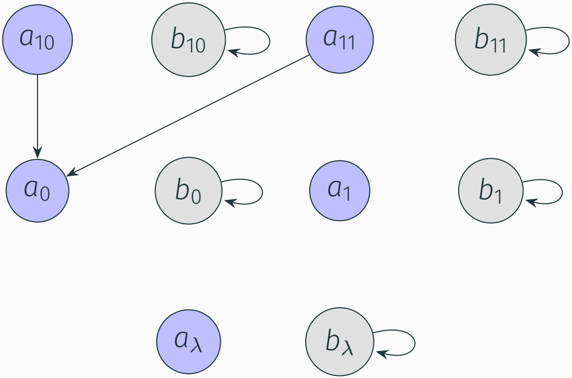
Enconding the tree $\{\lambda, 0, 1, 10, 11\}$ into an AF:



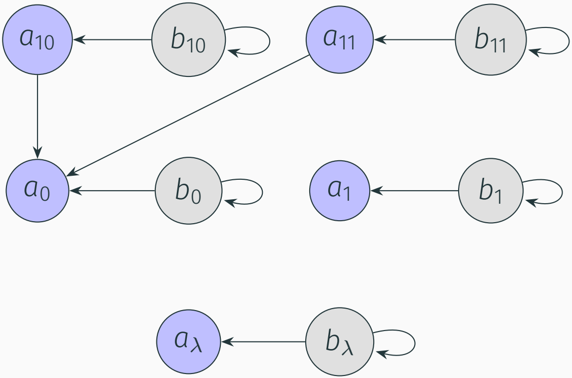
Enconding the tree $\{\lambda, 0, 1, 10, 11\}$ into an AF:



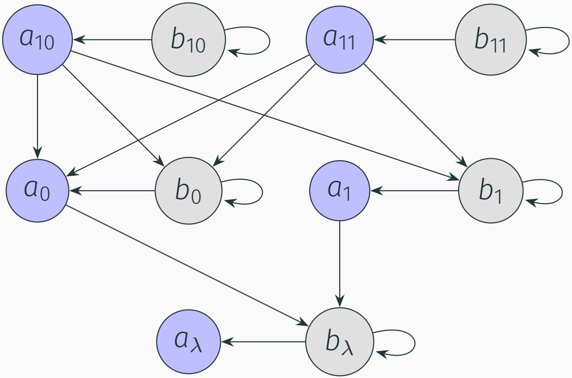
Enconding the tree $\{\lambda, 0, 1, 10, 11\}$ into an AF:



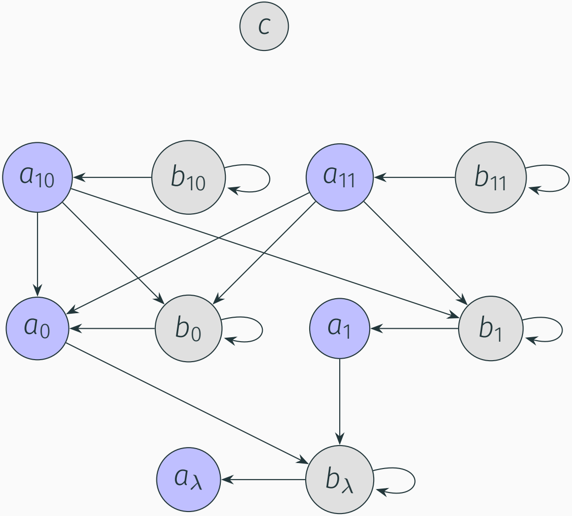
Enconding the tree $\{\lambda, 0, 1, 10, 11\}$ into an AF:



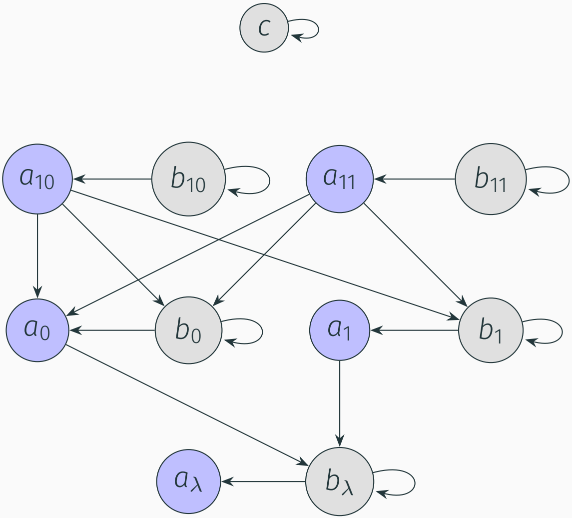
Enconding the tree $\{\lambda, 0, 1, 10, 11\}$ into an AF:



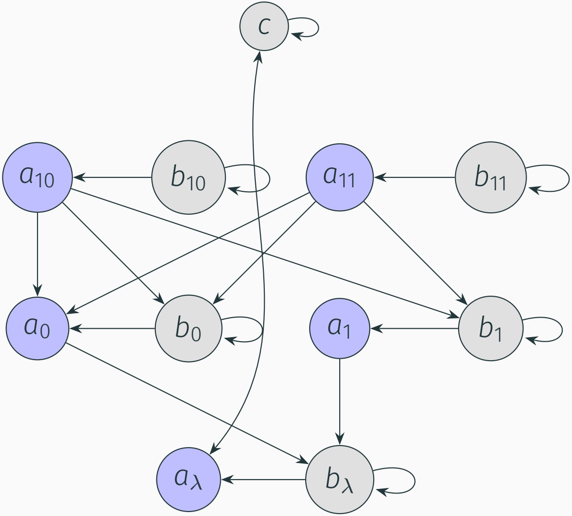
Enconding the tree $\{\lambda, 0, 1, 10, 11\}$ into an AF:



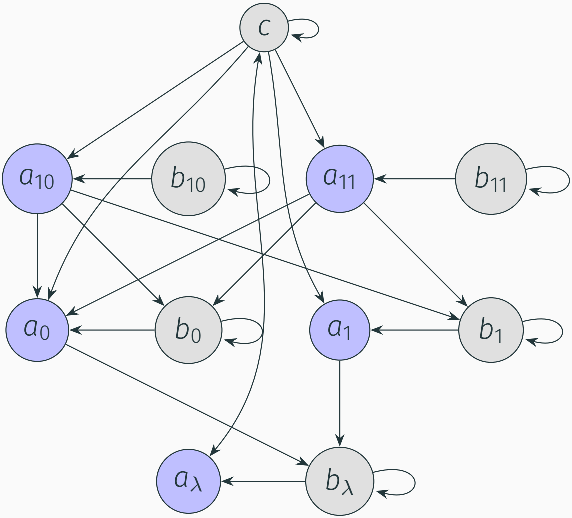
Enconding the tree $\{\lambda, 0, 1, 10, 11\}$ into an AF:



Enconding the tree $\{\lambda, 0, 1, 10, 11\}$ into an AF:



Enconding the tree $\{\lambda, 0, 1, 10, 11\}$ into an AF:



Characterizing extensions of $\mathcal{F}^{\mathcal{T}}$

Lemma: *A non-empty extension S of $\mathcal{F}^{\mathcal{T}}$ is admissible iff S is complete iff S is stable iff S is exactly $\{a_\sigma : \sigma \prec \pi\}$ for some π a path through $\mathcal{T}$.*

This construction shows all of the claimed hardness results (lower bounds of complexity). Let's see one.

Theorem (Andrews, S.)

$Cred_{ad}^{\infty}$ is Σ^1_1-complete.

Proof: For any tree $\mathcal{T}$, we produce $\mathcal{F}^{\mathcal{T}}$ and take e the index for $\mathcal{F}^{\mathcal{T}}$. Then $\langle e, a_\lambda \rangle \in \mathrm{Cred}_{ad}^{\infty}$ iff there is a path π through $\mathcal{T}$. Thus, we reduce the Σ^1_1-complete problem of determining whether a tree has a path to $\mathrm{Cred}_{ad}^{\infty}$.

Moral: Infinite AFs cannot be approximated

Let's conclude by highlighting an interesting byproduct of our theorems:

Those hesitant to venture into infinitary argumentation may suggest that any countably infinite argumentation framework (AF) $\mathcal{F}$ could be approximated by an increasing sequence of finite AFs $(\mathcal{F}_t)_{t\in\mathbb{N}}$, where each $\mathcal{F}_t$ represents the agent's knowledge at time t —after that, one may hope that analyzing such a sequence would provide enough insight about $\mathcal{F}$.

Nonetheless, a consequence of our findings is that, in general, there is absolutely no hope to understand acceptance of arguments in $\mathcal{F}$ in terms of some kind of limiting procedure studying the sequence $(\mathcal{F}_t)_{t\in\mathbb{N}}$.

Thank you!