On Computational Problems for Infinite Argumentation Frameworks: The Complexity of Finding Acceptable Extensions*

Uri Andrews[1,†], Luca San Mauro[2,*,†]

[1]Department of Mathematics, University of Wisconsin, USA
[2]Department of Philosophy, University of Bari, Italy

Abstract
This paper investigates infinite argumentation frameworks. We introduce computability theoretic machinery as a robust method of evaluating, in the infinite setting, the complexity of the main computational issues arising from admissible, complete, and stable semantics: in particular, for each of these semantics, we measure the complexity of credulous and skeptical acceptance of arguments, and that of determining existence and uniqueness of extensions. We also propose a way of using Turing degrees to classify, for a given infinite argumentation framework, the exact difficulty of computing an extension in a given semantics and show that these problems give rise to a rich class of complexities.

Keywords
infinite argumentation frameworks, computability theory, analytic and co-analytic sets, admissible extensions, stable extensions, complete extensions, Turing degrees

1. Introduction

Abstract argumentation theory is a fundamental research area in AI, providing a powerful paradigm for reasoning about knowledge representation and multi-agent systems. Historically, the focus has predominantly been on finite argumentation frameworks (AFs), leaving the infinite case relatively unexplored. As noted in [1], this oversight poses significant theoretical, conceptual, and practical limitations.

Firstly, infinite frameworks align naturally with Dung's seminal approach [2], whose results do not presuppose finiteness. Secondly, representing argumentation scenarios in an infinite manner captures the inherently nonmonotonic nature of argumentation, where arguments can always be challenged by the emergence of new information, making any fixed limit on the space of arguments somewhat artificial. Moreover, if one conceives an argumentative scenario with arguments being added as time proceeds, e.g., the collection of scientific studies, then infinite frameworks naturally emerge as the union of the argumentation frameworks that we see at each finite time. Thirdly, infinite AFs may arise in practical contexts, such as logic programming [3] and the logical analysis of multi-agent or distributed systems [4] (the substantial introduction of [1] provides other concrete examples of applications of infinite AFs, e.g., to multiagent negotiations).

Fortunately, recent years have seen a growing interest in infinite AFs, with special focus on how the existence and interplay of various semantics—well-understood for finite AFs—are affected in the infinite realm (see, e.g., [5, 6, 7, 8, 9]). This increasing recognition underscores the importance of infinite AFs for a broad understanding of argumentation theory.

However, the literature still lacks a comprehensive framework for systematically exploring all logical aspects of infinite AFs, particularly regarding their core computational issues. A significant research avenue in finite AFs has been determining the algorithmic complexity of tasks associated with finding coherent collections of arguments (up to suitable collection of semantics), with numerous complexity theoretic results highlighting their inherent computational intractability (see, e.g., [10, 11, 12]). To our knowledge, no analogous study has been conducted for infinite AFs.

This paper addresses this gap by initiating a systematic study of the complexity of computational problems in infinite AFs. For this endeavor, we bring into the subject of argumentation theory the machinery of computability theory, which may be regarded as an infinitary companion of computational complexity theory and abounds with concepts and hierarchies for measuring the complexity of computing and defining countably infinite objects, providing the appropriate machinery for this endeavor.

The application of computability theoretic tools out-

NMR 2024, 22nd International Workshop on Nonmonotonic Reasoning November 2-4, 2024, Hanoi, Vietnam

* A condensed version of this article was submitted to SAFA 2024 and an expanded version was submitted to FCR 2024.

* Corresponding author.
† These authors contributed equally.
andrews@math.wisc.edu (U. Andrews); luca.sanmauro@gmail.com (L. San Mauro)
https://math.wisc.edu/~andrews (U. Andrews); https://www.lucasanmauro.com/ (L. San Mauro)

side of mathematical logic is a well-established idea. Over the past decades, computability theory has been applied to a wide array of mathematical disciplines, and computability theoretic concepts have found applications in other formal subjects, such as theoretical computer science, economics, and linguistics (see, e.g., [13, 14, 15]).

The present paper, we argue, provides compelling evidence of the benefits of viewing infinite AFs through computability theoretic lenses. We assess the complexity of many computational problems—both established and novel—within our framework, illustrating their undecidability while providing precise measures of their complexity.

Organization of the paper

Section 2 briefly reviews the main semantic concepts from argumentation theory that are relevant to this paper, along with the fundamental computational problems associated with them. In Section 3, we introduce the key notions of computability theory employed in the work and we define the concept of computable AFs and the computational issues that emerges from it. Finally, in Sections 4 through 5, we determine the lower and upper bounds of the complexity for our computational problems: a critical technique for achieving hardness results involves suitably encoding trees into AFs. Our main results are collected in Tables 2 and 3.

2. Argumentation theoretic background

To keep our paper self-contained, we now briefly review some key concepts of Dung-style argumentation theory, focusing on the semantics notions considered in this paper and the fundamental computational problems associated with them (the surveys [16, 17] offer an overview of these topics).

An *argumentation framework* (AF) $\mathcal{F}$ is a pair $(A_\mathcal{F}, R_\mathcal{F})$ consisting of a set $A_\mathcal{F}$ of arguments and an attack relation $R_\mathcal{F} \subseteq A_\mathcal{F} \times A_\mathcal{F}$. If some argument a attacks some argument b, we may write $a \rightarrowtail b$ instead of $(a, b) \in R_\mathcal{F}$. Collections of arguments $S \subseteq A_\mathcal{F}$ are called *extensions.* For an extension S, the symbols S^+ and S^- denote, respectively, the arguments that S attacks and the arguments that attack S:

$$S^+ = \{x : (\exists y \in S)(y \rightarrowtail x)\};$$
$$S^- = \{x : (\exists y \in S)(x \rightarrowtail y)\}.$$

S *defends* an argument a, if any argument that attacks a is attacked by some argument in S (i.e., $\{a\}^- \subseteq S^+$). The *characteristic function* of $\mathcal{F}$ is the following mapping $f_\mathcal{F}$ which sends subsets of $A_\mathcal{F}$ to subsets of $A_\mathcal{F}$:

$$f_\mathcal{F}(S) := \{x : \ x \text{ is defended by } S\}.$$

All AFs investigated in this paper are infinite.

A *semantics* σ assigns to every AF $\mathcal{F}$ a set of extensions $\sigma(\mathcal{F})$ which are deemed as acceptable. A huge number of semantics, fueled by different motivations, have been proposed and analyzed. Here, we focus on three prominent choices, whose computational aspects are well-understood in the finite setting: admissible, complete, and stable semantics (abbreviated by *ad*, *stb*, *co*, respectively).

Let $\mathcal{F} = (A_\mathcal{F}, R_\mathcal{F})$ be an AF. Denote by $\mathit{cf}(\mathcal{F})$ the collection of extensions of $\mathcal{F}$ which are *conflict-free* (i.e., $S \in \mathit{cf}(\mathcal{F})$ iff $a \not\rightarrowtail b$, for all $a, b \in S$). Then, for $S \in \mathit{cf}(\mathcal{F})$,

- $S \in \mathit{ad}(\mathcal{F})$ iff S is self-defending (i.e., $S \subseteq f_\mathcal{F}(S)$);
- $S \in \mathit{co}(\mathcal{F})$ iff S is a fixed point of f_F (i.e., $S = f_\mathcal{F}(S)$);
- $S \in \mathit{stb}(\mathcal{F})$, iff S attacks all arguments outside of it (i.e., $S^+ = A_\mathcal{F} \smallsetminus S$).

In discussing the complete extensions, we will also briefly mention the grounded extension, which is the unique smallest fixed point of f_F; in any AF, the grounded extension always exists [2, Theorem 3].

For a given semantics σ, the following are some well-known computational problems related to σ:

- Cred_σ (for *credulous* acceptance) is the decision problem whose accepting instances are the pairs $(\mathcal{F}, a)$ so that $a \in S$ for *some* $S \in \sigma(\mathcal{F})$;
- Skept_σ (for *skeptical* acceptance) is the decision problem whose accepting instances are the pairs $(\mathcal{F}, a)$ so that $a \in S$ for *all* $S \in \sigma(\mathcal{F})$;
- Exist_σ is the decision problem whose accepting instances are the AFs $\mathcal{F}$ so that $\sigma(\mathcal{F}) \neq \emptyset$;
- NE_σ is the decision problem whose instances are the AFs $\mathcal{F}$ so that $\sigma(\mathcal{F}) \smallsetminus \{\emptyset\} \neq \emptyset$;
- Uni_σ is the decision problem whose accepting instances are the AFs $\mathcal{F}$ so that $|\sigma(\mathcal{F})| = 1$.

In formal argumentation theory, evaluating the computational complexity of the aforementioned problems for various semantics has been a noteworthy research thread for more than 20 years[17]. Table 1 collects known complexity results for the admissible, stable, and complete semantics. This analysis refers only to finite AFs. In the next section, we introduce our computability theoretic perspective that allows us to tackle complexity issues concerning infinite AFs.

σ	Cred_σ	Skept_σ	Exist_σ	NE_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

Table 1
Computational problems for finite AFs. $\mathcal{C}$-c denotes completeness for the class $\mathcal{C}$.

3. Computational problems for AFs through the lens of computability theory

In this section, we introduce computable AFs and we revisit the computational problems of the last section through the lens of computability theory. We aim at conveying the main ideas without delving into too many technical details. A more formal and comprehensive exposition of the fundamentals of computability theory can be found, e.g., in the textbooks [18]. We begin by establishing standard notation and terminology for some combinatorial notions that appear frequently in our proofs.

3.1. Sequences, strings, and trees

As is common in computability theory, we denote the set of natural numbers by ω. Since there is no risk of ambiguity, we simply refer to the elements of ω as numbers. The symbol ω^ω denotes the set of all functions from ω to ω. For our purposes, it is convenient to represent elements of ω^ω as infinite sequences of numbers (where the $i+1$th bit of $\pi \in \omega^\omega$ will be the output of the function π on input i). We denote by 0^∞ the infinite sequence consisting of only 0's (or, equivalently, the constant function to 0). The restriction of an infinite sequence $\pi \in \omega^\omega$ to its first n bits is denoted by $\pi \restriction_n$.

We use standard notation and terminology about *strings*: The set of all finite strings of numbers is denoted by $\omega^{<\omega}$. The symbol λ denotes the empty string. The concatenation of strings σ, τ is denoted by $\sigma^\frown\tau$. The length of a string σ is denoted by $|\sigma|$. If there is ρ so that $\sigma^\frown\rho = \tau$, we say that σ is a *prefix* of τ and we write $\sigma \preceq \tau$. Similarly, if $\pi \in \omega^\omega$ and $\sigma = \pi \restriction_n$ for some n, we write $\sigma \prec \pi$.

In order to formulate our problems as subsets of ω, it will be convenient to encode pairs of numbers into single numbers. The pairing function does this. Fix $p : \omega \times \omega \to \omega$ to be a computable bijection. We adopt the common habit of denoting $p(x, y)$ by $\langle x, y\rangle$.

The encodings discussed in Section 4 heavily rely on the difficulty of calculating paths through trees. As is common in computability theory, we say that a *tree* is a set $\mathcal{T} \subseteq \omega^{<\omega}$ closed under prefixes. We picture trees growing upwards, with $\sigma^\frown i$ to the left of $\sigma^\frown j$, whenever $i < j$. A *path* $\pi \in \omega^\omega$ through a tree $\mathcal{T} \subseteq \omega^{<\omega}$ is an infinite sequence so that $\pi \restriction_n \in \mathcal{T}$, for all numbers n. The set of paths through a tree $\mathcal{T}$ is denoted by $[\mathcal{T}]$. $\mathcal{T}$ is *well-founded* if $[\mathcal{T}] = \emptyset$ and otherwise is *ill-founded.* Note that we follow the standard terminology in computability theory requiring that paths be infinite. Indeed, if one were to allow paths to be finite, then these notions trivialize, since one could computably find a path through any given computable tree. For example, the set of strings

$$\mathcal{T} := \{\lambda\} \cup \{\sigma, \sigma^\frown 1 : (\forall n < |\sigma|)(\sigma(n) = 0)\}$$

is an ill-founded tree with $[\mathcal{T}] = \{0^\infty\}$.

If $\mathcal{T}$ contains strings of arbitrary length, then $\mathcal{T}$ has *infinite height.* Note that there are trees of infinite height which are well-founded, e.g., $\mathcal{T} = \{\lambda\} \cup \{n^\frown\sigma : |\sigma| \leq n\}$.

3.2. Computable argumentation frameworks

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. 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.

Notation. Let $(\Phi_e)_{e\in\omega}$ be a uniform enumeration of all partial computable functions from ω to $\{0, 1\}$.

Definition 3.1. *A number* e *is a* computable index for an AF $\mathcal{F} = (A_\mathcal{F}, R_\mathcal{F})$ *with* $A_\mathcal{F} = \{a_n : n \in \omega\}$ *so that*

$$\Phi_e(\langle n, m\rangle) = \begin{cases} 1 & \textit{if } a_n \rightarrowtail a_m \\ 0 & \textit{otherwise.} \end{cases}$$

An AF $\mathcal{F}$ *is* computable*, if it has a computable index* $e \in \omega$.

We use the notation $\mathcal{F}_e$ to refer to the AF with computable index e (note that every computable AF possesses infinitely many computable indices.).

Remark 3.2. *The collection of computable indices for AFs just defined is noncomputable (in particular, any index* e *for a non-total computable function* Φ_e *cannot be a computable index for an argumentation framework). There are alternative indexings that circumvent this issue; yet, adopting another indexing would not alter the complexity of the computational problems we analyze, though it would make the proofs slightly more cumbersome. Hence, we opt for the simplicity of Definition 3.1.*

The benefit of dealing with computable AFs is that the complexity of the decision problems associated with them do not arise due to complexity of the argumentation framework itself, but rather reflects the inherent complexity of the decision problem. Further, the computational problems associated with computable AFs can be naturally represented as subsets of ω, which are suitable to be classified by computability theoretic means:

Definition 3.3. *For a semantics* σ*:*

1. $Cred_\sigma^\infty := \{\langle e, n\rangle : (\exists S \in \sigma(\mathcal{F}_e))(a_n \in S)\}$;
2. $Skept_\sigma^\infty := \{\langle e, n\rangle : (\forall S \in \sigma(\mathcal{F}_e))(a_n \in S)\}$;
3. $Exist_\sigma^\infty := \{e : (\exists S \subseteq A_{\mathcal{F}_e}))(S \in \sigma(\mathcal{F}_e))\}$;
4. $NE_\sigma^\infty := \{e : (\exists S \in \sigma(\mathcal{F}_e))(S \neq \emptyset)\}$;
5. $Uni_\sigma^\infty := \{e : (\exists! S \subseteq A_{\mathcal{F}_e})(S \in \sigma(\mathcal{F}_e))\}$.

We also introduce new semantics which make sense only in the infinite setting. This is motivated by the idea that, given an infinite AF, we might hope for our accepted sets to give us infinitely much information.

1. $S \in \mathit{infad}(\mathcal{F})$ if and only if $S \in \mathit{ad}(\mathcal{F})$ and S is infinite;
2. $S \in \mathit{infco}(\mathcal{F})$ if and only if $S \in \mathit{co}(\mathcal{F})$ and S is infinite;
3. $S \in \mathit{infstb}(\mathcal{F})$ if and only if $S \in \mathit{stb}(\mathcal{F})$ and S is infinite.

As an illustration of why we might want to accept only infinite extensions, we consider that a given infinite AF may contain a single argument b so that b attacks every other argument, and every other argument attacks b. We imagine that b is a statement of extreme solipsism denying the truth of any other statement. While $\{b\}$ is a stable extension, it represents a negligible fraction of arguments, and we may prefer not to accept it. In an infinite AF, any finite set is as negligible as $\{b\}$, so we may prefer to accept only infinite extensions.

The complexity classes that most naturally match the problems of Definition 3.3 are those of the Σ_1^1 and Π_1^1 sets. The Σ_1^1 sets are formally defined as those subsets of ω that are definable in the language of second-order arithmetic using a single second-order existential quantifier ranging over subsets of ω followed by number quantifiers and the first order functions and relations $(+, \cdot, <, 0, 1, \in)$; for more details, see [18, §16]. Π_1^1 sets are the complements of Σ_1^1 sets.

Proposition 3.4. $Cred_\sigma^\infty$*,* $Exist_\sigma^\infty$*, and* NE_σ^∞ *are* Σ_1^1*, for* $\sigma \in \{\mathit{ad}, \mathit{stb}, \mathit{co}, \mathit{infad}, \mathit{infstb}, \mathit{infco}\}$.

Proof. We first consider $\sigma \in \{\mathit{ad}, \mathit{stb}, \mathit{co}\}$. To define $\mathrm{Cred}_\sigma^\infty$, we see from Definition 3.3: $\mathrm{Cred}_\sigma^\infty := \{\langle e, n\rangle \in \omega : (\exists S \in \sigma(\mathcal{F}_e)(a_n \in S))\}$ uses a single existential quantifier over sets S. This is similarly true for the definitions of $\mathrm{Exist}_\sigma^\infty$ and $\mathrm{NE}_\sigma^\infty$ in Definition 3.3. Thus, it suffices to see that the condition $S \in \sigma(\mathcal{F}_e)$ can be defined with only quantification over arguments, which are in bijection with ω, not needing quantification over sets of arguments.

Note that the definition of S^+ and S^- use only quantifiers over arguments. Thus the definition of $f_\mathcal{F}(S)$ given by $a \in f_\mathcal{F}(S)$ if and only if $\{a\}^- \subseteq S^+$ uses only quantifiers over arguments. Finally, $S \in \mathit{ad}(\mathcal{F}), S \in \mathit{stb}(\mathcal{F}), S \in \mathit{co}(\mathcal{F})$ are all defined from $f_\mathcal{F}(S)$ and S^+ using only quantifiers over arguments.

In the case of $\sigma \in \{\mathit{infad}, \mathit{infstb}, \mathit{infco}\}$, we need to also observe that S being infinite is defined via $\forall n \exists m (a_m \in S \wedge m > n)$, which uses only quantifiers over numbers. □

Proposition 3.5. $Skept_\sigma^\infty$ *is* Π_1^1*, whenever* σ *is in* $\{\mathit{ad}, \mathit{stb}, \mathit{co}, \mathit{infad}, \mathit{infstb}, \mathit{infco}\}$*. Furthermore, for* $\sigma \in \{\mathit{ad}, \mathit{co}\}$*,* Uni_σ^∞ *is* Π_1^1.

Proof. The definition of $\mathrm{Skept}_\sigma^\infty$ in Definition 3.3 uses a single universal set-quantifier followed by only number quantifiers in the definition of $\sigma(\mathcal{F}_e)$.

For $\sigma \in \{\mathit{ad}, \mathit{co}\}$, $e \in \mathrm{Uni}_\sigma^\infty$ if and only if there are not two different σ extensions (as there is always at least one σ extension). This is defined by the negation of the following formula:

$$(\exists S_1 \exists S_2)(\exists x \in S_1 \setminus S_2 \wedge S_1 \in \sigma(\mathcal{F}_e) \wedge S_2 \in \sigma(\mathcal{F}_e)).$$

Note that $\exists S_1 \exists S_2$ can be replaced by a single existential quantifier by encoding the pair (S_1, S_2) as a single set $\{\langle 1, x\rangle : x \in S_1\} \cup \{\langle 2, y\rangle : y \in S_2\}$. This shows that $\mathrm{Uni}_\sigma^\infty$ is the complement of a Σ_1^1 set, thus is Π_1^1. □

Remark 3.6. The above argument does not suffice to show that $\mathrm{Uni}_{\mathit{stb}}^\infty$ is also Π_1^1, since some AFs have no stable extension. The most obvious definition says there exists one stable extension and there does not exist two, which gives a definition which is a conjunction of a Σ_1^1 and a Π_1^1 condition, i.e., a so-called d-Σ_1^1 definition. This is analogous to the fact that in the finite case $\mathrm{Uni}_{\mathit{stb}}$ is **DP**-complete. Similarly, the argument above does not show that $\mathrm{Uni}_\sigma^\infty$ is Π_1^1 for $\sigma \in \{\mathit{infad}, \mathit{infstb}, \mathit{infco}\}$. Yet, it is true that $\mathrm{Uni}_\sigma^\infty$ is Π_1^1 for $\sigma \in \{\mathit{stb}, \mathit{infad}, \mathit{infstb}, \mathit{infco}\}$ as we show below in Corollaries 5.5,5.10, and 5.14.

We note that knowing that a problem is Σ_1^1 does not necessarily mean the problem is complicated. This only gives an upper-bound for its complexity. Sometimes, a simpler definition is achievable. As an example, we consider $\mathrm{Cred}_{\mathit{cf}} := \{\langle e, n\rangle : (\exists S \in \mathit{cf}(\mathcal{F}_e))(a_n \in S)\}$.

Though the given definition is Σ_1^1, to know if an argument a_n belongs to a conflict-free extension of $\mathcal{F}_e$, it suffices to check whether a_n is non-self-defeating, i.e., $a_n \not\rightarrow a_n$, which is equivalent to checking the computable fact that $\Phi_e(\langle n, n\rangle) = 0$. In contrast, we will show that for the computational problems associated to the admissible, stable, and complete semantics, the use of the quantifier ranging over all sets cannot be avoided.

We will heavily rely on the following fundamental theorem by Kleene which offers a natural way of representing Σ_1^1 sets:

Theorem 3.7 (Kleene [19]). *A set $X \subseteq \omega$ is Σ_1^1 if and only if there is a computable sequence of computable trees $(\mathcal{T}_n^X)_{n\in\omega}$ so that $n \in X$ iff $\mathcal{T}_n^X$ is ill-founded.*

We call $(\mathcal{T}_n^X)_{n\in\omega}$ a *tree-sequence* for X. As a corollary of Kleene's theorem, one obtains that the problem of deciding which computable trees in $\omega^{<\omega}$ are ill-founded (or well-founded) is as hard as any other Σ_1^1 (resp., Π_1^1) problem.

Theorem 3.7 gives a reason to consider the Σ_1^1 sets as the natural infinite analogs of the **NP** problems. Namely, given an ill-founded computable tree $\mathcal{T}$ and a sequence π which is a path through $\mathcal{T}$, it is relatively simple to check that $\pi \in [\mathcal{T}]$ (it requires checking infinitely many simple facts: $\pi \restriction_n \in \mathcal{T}$, for each n), but finding a sequence $\pi \in [T]$—or even knowing whether there exists a sequence $\pi \in [T]$—is a far harder problem.

Our main goal is to exactly characterize the complexity of the computational problems described in Definition 3.3. To do so, we need to show that they are complete for their respective complexity classes. The following definition formalizes this notion.

Definition 3.8. *Let Γ be a complexity class (e.g., $\Gamma \in \{\Sigma_1^1, \Pi_1^1\}$). A set $V \subseteq \omega$ is* Γ-hard, *if for every $X \in \Gamma$ there is a computable function $f : \omega \to \omega$ so that $x \in X$ if and only if $f(x) \in V$. If V is Γ-hard and belongs to Γ, then it is* Γ-complete.

Proposition 3.9. *It follows from Theorem 3.7 that the set of indices for ill-founded computable trees is a Σ_1^1-complete set. Similarly, the set of indices for well-founded computable trees is a Π_1^1-complete set.*

The following example is far less obvious, but will be useful below to examine Uni_σ^∞.

Theorem 3.10 ([20, Theorem 18.11]). *The set UB of indices for computable trees with exactly one path is a Π_1^1-complete set.*

Remark 3.11. The hardness in Theorem 3.10 is quite easy. We can reduce the question of whether a tree $\mathcal{T}$ is well-founded to whether a tree $\mathcal{T}'$ has two paths, where $\mathcal{T}'$ always has at least one path, by simply giving $\mathcal{T}'$ one more path than $\mathcal{T}$ (e.g. $\mathcal{T}' = \{1^\frown\sigma : \sigma \in \mathcal{T}\} \cup \{\sigma : (\forall n < |\sigma|)\sigma(n) = 0\}$). The fact that UB is itself Π_1^1 is the subtle part of this example.

Theorem 3.7 along with Definition 3.8 suggest a natural approach for gauging the complexity of the computational problems of Definition 3.3. Namely, given another Σ_1^1 (or Π_1^1) set X, we translate the question asking whether $n \in X$ to the question of if the tree $\mathcal{T}_n^X$ is ill-founded (resp., well-founded), and then we need to computably find an instance of our computational problem which should be accepted if and only if $\mathcal{T}_n^X$ is ill-founded (resp., well-founded). This involves coding a tree, or more precisely, the collection of paths through a tree into the σ extensions in an argumentation framework. We do exactly this in Section 4.

Table 2 collects our results regarding complexities of the computational problems examined for computable argumentation frameworks.

Remark 3.12. As noted before, the Σ_1^1 sets are natural analogs in the infinitary setting of the **NP** sets, and the Π_1^1 sets are the natural analogs of the **coNP** sets. With the exception of Skept_{co}^∞ and Uni_{stb}^∞, Table 2 follows this translation from Table 1 for the first three rows. These two results mark surprising differences in the infinite setting.

The trivial entries are due to the fact that $\emptyset$ is always an admissible extension and the grounded extension is always a complete extension.

3.3. Spectra of σ extensions

We propose a way to more fully understand the complexity of the problem of finding a σ extension in a given AF $\mathcal{F}$.

Definition 3.13. *For each $e \in \omega$ and semantics σ, let $Spec_\sigma^{\neg\emptyset}(\mathcal{F}_e)$ be the set of Turing degrees of non-empty sets $X \subseteq \omega$ so that $\{a_n : n \in X\}$ is a σ extension in $\mathcal{F}_e$.*

The notion $\text{Spec}_\sigma^{\neg\emptyset}(\mathcal{F}_e)$ exactly captures the difficulty of computing a non-empty σ extension in $\mathcal{F}_e$. We will be relating the problem of computing a σ extension in $\mathcal{F}_e$ to the problem of finding a path through a particular tree. So, we define the analogous notion of the spectrum of a tree.

Definition 3.14. *Given any computable tree $\mathcal{T}$, we let $Spec(\mathcal{T})$ be set of Turing degrees of paths $X \in [\mathcal{T}]$.*

Table 3 collects our results on spectra of extensions.

σ	$\mathsf{Cred}_\sigma^\infty$	$\mathsf{Skept}_\sigma^\infty$	$\mathsf{Exists}_\sigma^\infty$	$\mathsf{NE}_\sigma^\infty$	$\mathsf{Uni}_\sigma^\infty$
ad	Σ_1^1-c 4.2,3.4	trivial	trivial	Σ_1^1-c 4.2 3.4	Π_1^1-c 4.3,3.5
stb	Σ_1^1-c 4.2,3.4	Π_1^1-c 4.4,3.5	Σ_1^1-c 4.2, 3.4	Σ_1^1-c 4.2,3.4	Π_1^1-c 4.3, 5.10
co	Σ_1^1-c 4.2, 3.4	Π_1^1-c $*$, 3.5	trivial	Σ_1^1-c 4.2, 3.4	Π_1^1-c 4.3,3.5
infad	Σ_1^1-c 4.2,3.4	Π_1^1-c 4.4,3.5	Σ_1^1-c 4.2, 3.4	Σ_1^1-c 4.2, 3.4	Π_1^1-c 4.3,5.5
infstb	Σ_1^1-c 4.2,3.4	Π_1^1-c 4.4,3.5	Σ_1^1-c 4.2, 3.4	Σ_1^1-c 4.2, 3.4	Π_1^1-c 4.3,5.10
infco	Σ_1^1-c 4.2,3.4	Π_1^1-c 4.4,3.5	Σ_1^1-c 4.2, 3.4	Σ_1^1-c 4.2, 3.4	Π_1^1-c 4.3, 5.14

Table 2
Computational problems for computable AFs. $\mathcal{C}$-c denotes completeness for the class $\mathcal{C}$. The entry with an asterisk is not fully proved in this paper. Rather, the Π_1^1-hardness for $\mathsf{Skept}_{co}^\infty$ is deferred to future work focusing on the grounded semantics. It is included in the table here (though partially unproved) to give a more complete picture. The numbers in each cell of the table refer to the Theorem number providing the lower bound and upper bounds for the result in that cell.

σ	$\mathsf{Spec}_\sigma^{\neg\emptyset}$
ad	Exactly Spec($\mathcal{T}$)
stb	Exactly Spec($\mathcal{T}$)
co	Any Spec($\mathcal{T}$)
infad	Exactly Spec($\mathcal{T}$)
infstb	Exactly Spec($\mathcal{T}$)
infco	Any Spec($\mathcal{T}$)

Table 3
For any computable tree $\mathcal{T}$, there is a computable argumentation framework $\mathcal{F}_e$ so that $\mathsf{Spec}(\mathcal{T}) = \mathsf{Spec}_\sigma^{\neg\emptyset}(\mathcal{F}_e)$. When $\sigma \in \{ad, stb, infad, infstb\}$, the converse also holds. Namely, for every e, there is a computable tree so that $\mathsf{Spec}_\sigma^{\neg\emptyset}(\mathcal{F}_e) = \mathsf{Spec}(\mathcal{T})$. We do not know how to attain a corresponding upper bound for the complete or infinite complete cases.

4. Encoding a tree into an argumentation framework

Given a tree $\mathcal{T} \subseteq \omega^{<\omega}$, we will define an argumentation framework $\mathcal{F}^\mathcal{T} = (A^\mathcal{T}, R^\mathcal{T})$. 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 $\tau \not\prec \sigma$;
5. $c \rightarrowtail a_\tau$ for every $\tau \in \mathcal{T}$;
6. $a_\lambda \rightarrowtail c$.

Figure 1 gives an example of our encoding for a finite tree. We next consider which extensions in $\mathcal{F}^\mathcal{T}$ are admissible, stable, or complete.

Notation. For $\pi \in [\mathcal{T}]$, let S_π be the set $\{a_\sigma : \sigma \prec \pi\}$.

Lemma 4.1. *A non-empty extension S of $\mathcal{F}^\mathcal{T}$ is stable iff S is complete iff S is admissible iff S is exactly S_π for some $\pi \in [\mathcal{T}]$.*

Proof. Stable always implies complete, which always implies admissible. It is straightforward to check that S_π is stable for any $\pi \in [\mathcal{T}]$, so we need only show that any non-empty admissible extension is exactly some S_π. Suppose that S is admissible. Observe that c and b_σ cannot be in S since these are self-defeating. So some $a_\sigma \in S$ since S is non-empty. Note that since $c \rightarrowtail a_\sigma$, we must have $a_\lambda \in S$. Next, observe that if $a_\tau \in S$, then there must be some i so that $a_{\tau\frown i} \in S$: this is because some element of S must defend a_τ from b_τ and such an element must be an a_σ with $|\sigma| = |\tau| + 1$. But it must have $\tau \prec \sigma$ as otherwise a_σ would attack a_τ. It follows that S contains S_π for some $\pi \in [\mathcal{T}]$. Since S_π is stable, S cannot properly contain S_π, so $S = S_\pi$. ☐

We are now in a position to obtain hardness results for the computational problems described in Definition 3.3.

Theorem 4.2. *The following hold:*

1. *for $\sigma \in \{ad, stb, co, infad, infstb, infco\}$, NE_σ^∞ is Σ_1^1-hard;*
2. *for $\sigma \in \{stb, infad, infstb, infco\}$, $Exist_\sigma^\infty$ is Σ_1^1-hard;*
3. *for $\sigma \in \{ad, stb, co, infad, infstb, infco\}$, $Cred_\sigma^\infty$ is Σ_1^1-hard.*

Proof. 1. Let $X \in \Sigma_1^1$ and let $(\mathcal{T}_n^X)_{n\in\omega}$ be a tree-sequence for X, as given by Theorem 3.7. To show Σ_1^1-hardness, we need to produce a computable function f so that $n \in X$ if and only if $f(n) \in \mathsf{NE}_\sigma^\infty$. We let $f(n)$ be a computable index for $\mathcal{F}^{\mathcal{T}_n^X}$. Then Lemma 4.1 shows that $n \in X$ if and only if $\mathcal{T}_n^X$ is ill-founded if and only if $\mathcal{F}^{\mathcal{T}_n^X}$ has a non-empty σ extension for each $\sigma \in \{ad, stb, co, infad, infstb, infco\}$.

2. For each of these σ, the empty set is not a σ extension, so $\mathsf{Exist}_\sigma^\infty = \mathsf{NE}_\sigma^\infty$, which we showed above is Σ_1^1-hard.

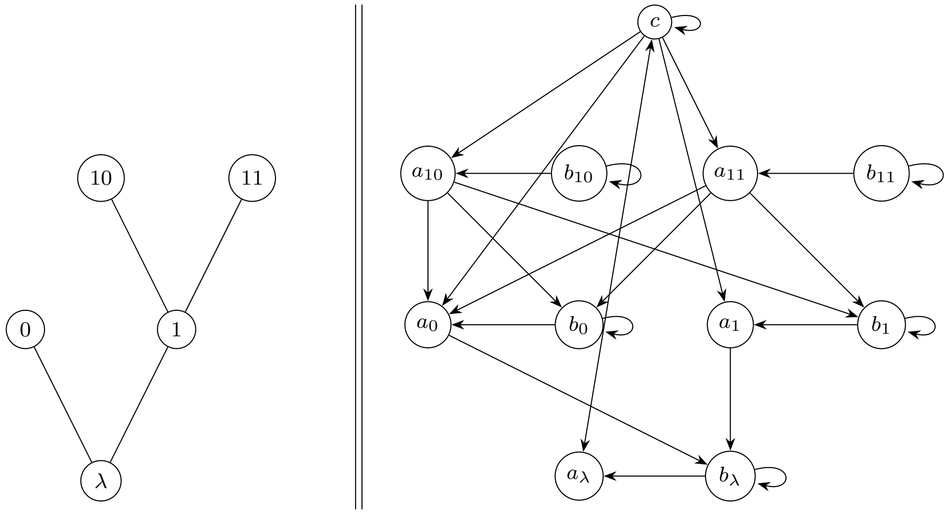

Figure 1: Example of our encoding of trees into AFs: the left-side represents the tree $\{\lambda, 0, 1, 10, 11\}$, the right-side is the resulting AF. When applied to trees T of infinite height, $[T]$ will be encoded into the stable, complete, and admissible extensions of $\mathcal{F}_T$. For the example shown in the figure, the only admissible extension of $\mathcal{F}_{\mathcal{T}}$ is the empty one, since $[T] = \emptyset$.

3. In the proof of 1. above, we reduced a given Σ^1_1 set X to $\mathrm{NE}^\infty_\sigma$ by sending n to $\mathcal{F}^{\mathcal{T}^X_n}$. Note that $\mathcal{F}^{\mathcal{T}^X_n}$ has a non-empty σ extension if and only if a_λ is in some σ extension. Thus sending n to $\langle e, a_\lambda\rangle$ where e is a computable index for $\mathcal{F}^{\mathcal{T}^X_n}$ shows that $\mathrm{Cred}^\infty_\sigma$ is Σ^1_1-hard. □

Theorem 4.3. *For $\sigma \in \{ad, stb, co, infad, infstb, infco\}$, Uni^∞_σ is Π^1_1-hard.*

Proof. We first consider $\sigma \in \{ad, co\}$. Let $X \in \Pi^1_1$ and let $(\mathcal{T}^{\omega\smallsetminus X}_n)_{n\in\omega}$ be a tree-sequence for its complement. Consider the sequence of AFs $(\mathcal{F}^{\mathcal{T}^{\omega\smallsetminus X}_n})_{n\in\omega}$. Note that $\emptyset$ is an admissible extension in any AF and since every argument in $\mathcal{F}^{\mathcal{T}^{\omega\smallsetminus X}_n}$ is attacked, $\emptyset$ is also a complete extension. Thus, $\mathcal{F}^{\mathcal{T}^{\omega\smallsetminus X}_n}$ has a unique σ extension if and only if $\mathcal{T}^{\omega\smallsetminus X}_n$ is well founded if and only if $n \in X$, which shows that Uni^∞_{ad} and Uni^∞_{co} are Π^1_1-hard.

For the other σ, $\emptyset$ is not a σ extension. We use Theorem 3.10 to show Π^1_1-hardness. Let X be any Π^1_1 set. Then we get from Remark 3.11 a sequence of trees $\mathcal{T}'_n$ so that $0^\infty \in [\mathcal{T}'_n]$ for each n, and $\{n : \mathcal{T}'_n \text{ has only one path}\}$ is Π^1_1-hard. It follows from Lemma 4.1 that this holds if and only if $\mathcal{F}^{\mathcal{T}'_n}$ has a unique σ extension, which shows the Π^1_1-hardness of $\mathrm{Uni}^\infty_\sigma$. □

Theorem 4.4. *For any $\sigma \in \{stb, infstb, infco, infad\}$, $Skept^\infty_\sigma$ is Π^1_1-hard.*

Proof. Let X be a Π^1_1 set. Then we get from Remark 3.11 a sequence of trees $\mathcal{T}'_n$ so that $0^\infty \in [\mathcal{T}'_n]$ for each n, and $\{n : \mathcal{T}'_n \text{ has only one path}\}$ is Π^1_1-hard. Then note that $\langle e, a_0\rangle \in \mathrm{Skept}^\infty_\sigma$ where e is a computable index for $\mathcal{G}_n := \mathcal{F}_{\mathcal{T}'_n}$ if and only if $\mathcal{T}'_n$ only has paths π with $\pi(0) = 0$ if and only if $\mathcal{T}'_n$ has only one path (see the definition of $\mathcal{T}'_n$ in Remark 3.11) if and only if $n \in X$. This shows the Π^1_1-hardness of $\mathrm{Skept}^\infty_\sigma$. □

Theorem 4.5. *For $\sigma \in \{ad, stb, co, infad, infstb, infco\}$ and for any computable tree $\mathcal{T}$, there exists a computable AF $\mathcal{F}_e$ so that $Spec^{\neg\emptyset}_\sigma(\mathcal{F}_e) = Spec(\mathcal{T})$.*

Proof. Observe that for the AF $\mathcal{F}_e = \mathcal{F}^{\mathcal{T}}$, it follows from Lemma 4.1 that the non-empty σ extensions are all infinite and are in the same Turing degrees as the paths through $\mathcal{T}$. □

5. Trees coding extensions in $\sigma(\mathcal{F})$

In this section, we give upper bounds to the complexity of $\mathrm{Spec}^{\neg\emptyset}_\sigma$ for $\sigma \in \{ad, stb, infad, infstb\}$ as well as giving upper bounds for the complexity of $\mathrm{Uni}^\infty_\sigma$ for any $\sigma \in \{stb, infad, infstb, infco\}$. We do this by describing how to computably encode the collection of extensions in $\sigma(\mathcal{F})$ into the set of paths through a tree.

The admissible case

Given a computable argumentation framework $\mathcal{F}$, we will describe a computable tree $\mathcal{T}^{\mathcal{F}}$ so that the paths of $\mathcal{T}^{\mathcal{F}}$ encode the non-empty admissible extensions in $\mathcal{F}$. We begin with an intuitive description of how a path π through the tree $\mathcal{T}^{\mathcal{F}}$ will encode an admissible extension S, and we give the formal definition of $\mathcal{T}^{\mathcal{F}}$ below.

Branching in $\mathcal{T}^{\mathcal{F}}$ will come in three flavors. The first branching is used to give the least element of the admissible extension S. This is to ensure that the extension is non-empty. If we wished to allow the empty extension, we could omit this branching. For any $j > 0$, the branching on level $2j$ serves to code whether or not $j \in S$. Branching on the odd levels serve to explain how S satisfies the hypothesis of being an admissible extension. If $a_i \rightarrowtail a_j$ is the nth element of some computable enumeration of all attacking pairs of arguments, then $\sigma(2n+1)$ will be 0 if $a_j \notin S$ and otherwise will be $k+1$ where k is least so that $a_k \in S$ and $a_k \rightarrowtail a_i$.

Let $\mathcal{F} = (A_{\mathcal{F}}, R_{\mathcal{F}})$ be a computable AF. Let $(g_n)_{n\in\omega}$ be a computable sequence of all elements of $R_{\mathcal{F}}$. If $g_n = (a_i, a_j)$, we denote a_i by g_n^- and a_j by g_n^+. We now define the tree $\mathcal{T}^{\mathcal{F}}$.

Definition 5.1. *Any given string $\sigma \in \omega^{<\omega}$ defines two subsets of arguments in $A_{\mathcal{F}}$:*

- $In_\sigma = \{a_{\sigma(0)}\} \cup \{a_j : j > 0 \wedge \sigma(2j) = 1\} \cup \{a_k : (\exists j)(\sigma(2j+1) = k+1)\} \cup \{a_i : (\exists j)(\sigma(2j+1) > 0 \wedge g_j^+ = a_i)\}$;
- $Out_\sigma = \{a_i : i < \sigma(0)\} \cup \{a_j : j > 0 \wedge \sigma(2j) = 0\} \cup \{a_i : (\exists j)(\sigma(2j+1) = 0 \wedge g_j^+ = a_i)\} \cup \{a_i : (\exists j)(\sigma(2j+1) > i+1 \wedge a_i \rightarrowtail g_j^-)\}$.

We define $\mathcal{T}^{\mathcal{F}}$ as the set of σ so that

- *In_σ is conflict-free;*
- *$In_\sigma \cap Out_\sigma = \emptyset$*
- *If $0 < 2j < |\sigma|$, then $\sigma(2j) \in \{0, 1\}$;*
- *If $2j+1 < |\sigma|$ and $\sigma(2j+1) = k+1$, then $a_k \rightarrowtail g_j^-$.*

Theorem 5.2. *Let $\mathcal{F}$ be a (computable) argumentation framework. Then the non-empty admissible extensions of $\mathcal{F}$ are in (computable) bijection with the paths in $\mathcal{T}^{\mathcal{F}}$.*

Proof. Given a non-empty admissible extension S of $\mathcal{F}$, we define the corresponding path π in $\mathcal{T}^{\mathcal{F}}$ as follows. Let $\pi(0)$ be the least element of S. For $j > 0$, let $\pi(2j) = 1$ if $a_j \in S$ and $\pi(2j) = 0$ otherwise. Let $\pi(2n+1)$ be 0 if $g_n^+ \notin S$ and be $k+1$ where k is least so that $a_k \in S$ and $a_k \rightarrowtail g_n^-$ otherwise. It is straightforward to check that $\pi \in [\mathcal{T}^{\mathcal{F}}]$.

Given a path π through $\mathcal{T}^{\mathcal{F}}$, first note that whenever there is some $\sigma \prec \pi$ so that $a_n \in In_\sigma$, then $\pi(2n) = 1$. This is because whenever $\sigma \preceq \tau$, then $In_\sigma \subseteq In_\tau$. Then since $\tau := \pi \restriction_{\max(|\sigma|, 2n+1)}$ is in $\mathcal{T}^{\mathcal{F}}$, we cannot have $\tau(2n) = 0$ since $a_n \in In_\tau$. Thus $\pi(2n) = \tau(2n) = 1$. It follows that $\bigcup_{\sigma\prec\pi} In_\sigma = \{a_n : \pi(2n) = 1\}$. The same argument shows that $\bigcup_{\sigma\prec\pi} Out_\sigma = \{a_n : \pi(2n) = 0\}$. Let $S = \{a_n : \pi(2n) = 1\}$.

Note that S is conflict-free, since if $a_i, a_j \in S$ then there is some long enough $\sigma \prec \pi$ so that $a_i, a_j \in In_\sigma$. Since $\sigma \in \mathcal{T}^{\mathcal{F}}$, it follows that In_σ is conflict-free, so $a_i \not\rightarrowtail a_j$. Next, observe that S defends itself. If $a_i \rightarrowtail a_j$ and $a_j \in S$, then there is some n so that $g_n = (a_i, a_j)$. Then consider $\sigma = \pi \restriction_{n+1}$. We must have $\sigma(n) = k+1$ for some k with $a_k \in S$ and $a_k \rightarrowtail a_i$.

Finally, note that both the map from π to S and from S to π are computable if $\mathcal{F}$ is computable and are inverses of each other. ☐

Corollary 5.3. *For every computable AF $\mathcal{F}_e$, there exists a computable tree $\mathcal{T}$ so that $Spec_{ad}^{\neg\emptyset}(\mathcal{F}_e) = Spec(\mathcal{T})$.*

Proof. It follows immediately from Theorem 5.2 that $\mathrm{Spec}_{ad}^{\neg\emptyset}(\mathcal{F}_e) = \mathrm{Spec}(\mathcal{T}^{\mathcal{F}_e})$. ☐

Corollary 5.4. *For every computable AF $\mathcal{F}_e$, there exists a computable tree $\widehat{\mathcal{T}}$ so that $Spec_{infad}^{\neg\emptyset}(\mathcal{F}_e) = Spec(\widehat{\mathcal{T}})$.*

Proof. The tree needed here is a slight alteration of the tree $\mathcal{T}^{\mathcal{F}}$. In $\mathcal{T}^{\mathcal{F}}$, we made $\sigma(0)$ tell us the least k so $a_k \in S$ so as to ensure that $S \neq \emptyset$. We do the same on infinitely many layers, e.g., instead of having $\sigma(2n)$ be 0 or 1 to tell us whether or not $n \in S$, we have $\sigma(2n)$ tell us the nth least number k so that $a_k \in S$. With the tree altered like this, paths are in computable bijection with the infinite admissible extensions. ☐

Corollary 5.5. *Uni_{infad}^{∞} is Π_1^1.*

Proof. e is in Uni_{infad} if and only if the tree $\widehat{\mathcal{T}}$ in Corollary 5.4 has a unique path. By Theorem 3.10, this is a Π_1^1 condition. ☐

The stable case

Similarly, we can construct a tree encoding the stable extensions by making $\sigma(n) = 0$ if $a_n \in S$ and otherwise making $\sigma(n)$ be $k+1$ where k is least so that $a_k \in S$ and $a_k \rightarrowtail a_n$.

Definition 5.6. *Any given string $\sigma \in \omega^{<\omega}$ defines two subsets of arguments in $A_{\mathcal{F}}$:*

- $In_\sigma = \{a_i : \sigma(i) = 0\} \cup \{a_{\sigma(i)-1} : i < |\sigma| \wedge \sigma(i) > 0\}$;
- $Out_\sigma = \{a_i : i < |\sigma| \wedge \sigma(i) > 0\} \cup \{a_i : (\exists j)\sigma(j) > i+1 \wedge a_i \rightarrowtail a_j)\}$.

We define $\mathcal{T}^{\mathcal{F}}$ as the set of σ so that

- *In_σ is conflict-free;*
- *$In_\sigma \cap Out_\sigma = \emptyset$*
- *If $j < |\sigma|$ and $\sigma(j) = k+1$, then $a_k \rightarrowtail a_j$.*

Theorem 5.7. *Let $\mathcal{F}$ be a (computable) argumentation framework. Then the stable extensions of $\mathcal{F}$ are in (computable) bijection with the paths in $\mathcal{T}^{\mathcal{F}}$.*

Proof. Given a stable extension S of $\mathcal{F}$, we define the corresponding path π in $\mathcal{T}^{\mathcal{F}}$ as follows. For each n, let $\pi(n)$ be 0 if $a_n \in S$ and let $\pi(n)$ be $k+1$ where k is least so that $a_k \in S$ and $a_k \rightarrowtail a_n$ otherwise. It is straightforward to check that $\pi \in [\mathcal{T}^{\mathcal{F}}]$.

Given a path π through $\mathcal{T}^{\mathcal{F}}$, first note that whenever there is some $\sigma \prec \pi$ so that $a_n \in In_\sigma$, then $\pi(n) = 0$. This is because whenever $\sigma \preceq \tau$, then $In_\sigma \subseteq In_\tau$. Then since $\tau = \pi \restriction_{\max(|\sigma|, n+1)}$ is on $\mathcal{T}^{\mathcal{F}}$, we cannot have $\tau(n) \neq 0$ since $a_n \in In_\tau$ and therefore cannot be in Out_τ. It follows that $\bigcup_{\sigma \prec \pi} In_\sigma = \{a_n : \pi(n) = 0\}$. Let $S = \{a_n : \pi(n) = 0\}$.

Note that S is conflict-free, since if $a_i, a_j \in S$, then $a_i \not\rightarrowtail a_j$ since $\sigma = \pi \restriction_{\max(i,j)+1}$ is in $\mathcal{T}^{\mathcal{F}}$, and thus In_σ is conflict-free. Next observe that for any n, either $a_n \in S$ or there is some m so that $a_m \in S$ and $a_m \rightarrowtail a_n$. In particular, if $\pi(n) = 0$, then $a_n \in S$ and otherwise $a_{\pi(n)-1} \in S$ and $a_{\pi(n)-1} \rightarrowtail a_n$.

Finally, note that both the map from σ to S and from S to σ are computable if $\mathcal{F}$ is computable and are inverses of each other. □

Corollary 5.8. *For any computable AF $\mathcal{F}_e$ there exists a computable tree $\mathcal{T}$ so that $Spec^{\neg\emptyset}_{stb}(\mathcal{F}_e) = Spec(\mathcal{T})$.*

Proof. This follows directly from Theorem 5.7 along with the fact that $\emptyset$ is never a stable extension. □

Corollary 5.9. *For any computable AF $\mathcal{F}_e$ there exists a computable tree $\mathcal{T}$ so that $Spec^{\neg\emptyset}_{infstb}(\mathcal{F}_e) = Spec(\mathcal{T})$.*

Proof. We add layers of branching to the tree as in Corollary 5.4 so that, e.g., $\sigma(2n) = m$ means that m is the nth least number so that $a_n \in S$. This produces a tree $\widehat{\mathcal{T}^{\mathcal{F}}}$ so that the paths are in computable bijection with the infinite stable extensions of $\mathcal{F}$. □

Corollary 5.10. *Uni^{∞}_{stb} and Uni^{∞}_{infstb} are Π^1_1.*

Proof. As the paths through $\mathcal{T}^{\mathcal{F}_e}$ are in bijection with the stable extensions, $e \in \mathrm{Uni}^{\infty}_{stb}$ if and only if $\mathcal{T}^{\mathcal{F}_e}$ has a unique path. As the paths through $\widehat{\mathcal{T}}^{\mathcal{F}}$ (see Corollary 5.9) are in bijection with the infinite stable extensions in $\mathcal{F}_e$, we have $e \in \mathrm{Uni}_{infstb}$ if and only if $\widehat{\mathcal{T}}^{\mathcal{F}_e}$ has a unique path. By Theorem 3.10, these are both Π^1_1 conditions. □

The complete case

Given an argumentation framework $\mathcal{F}$, we can similarly construct a tree $\mathcal{T}^{\mathcal{F}}$ so that paths through $\mathcal{T}^{\mathcal{F}}$ code complete extensions. In order to ensure that $f_{\mathcal{F}}(S) \subseteq S$, we will need the paths in $\mathcal{T}^{\mathcal{F}}$ to not only code sets S but also their attacked sets S^+.

Given an extension S, we will let $\pi \in \mathcal{T}^{\mathcal{F}}$ encode S and S^+ as follows:

- $\pi(2n) = 0$ if $a_n \in S$ and otherwise $\pi(2n) = k+1$ where k is least so $a_k \notin S^+$ and $a_k \rightarrowtail a_n$.
- $\pi(2n+1) = 0$ if $a_n \notin S^+$ and otherwise $\pi(2n+1) = k+1$ where k is least so $a_k \in S$ and $a_k \rightarrowtail a_n$.

Note that $\pi(2n)$ explains why a_n is either in S or it is not in $f_{\mathcal{F}}(S)$, i.e., $f_{\mathcal{F}}(S) \subseteq S$, while $\pi(2n+1)$ simply verifies that the elements which π says are in S^+ are in fact in S^+.

Formally, we define $\mathcal{T}^{\mathcal{F}}$ as follows:

Definition 5.11. *Any given string $\sigma \in \omega^{<\omega}$ defines four subsets of arguments in $A_{\mathcal{F}}$:*

- $In_\sigma = \{a_i : \sigma(2i) = 0\} \cup \{a_k : (\exists j)(\sigma(2j+1) = k+1\}$
- $Out_\sigma = \{a_i : \sigma(2i) \neq 0\} \cup \{a_i : (\exists j)\sigma(2j+1) > i+1 \wedge a_i \rightarrowtail a_j)\}$
- $InSplus_\sigma = \{a_i : (\exists j)(\sigma(2j) > i+1 \wedge a_i \rightarrowtail a_j)\} \cup \{a_i : \sigma(2i+1) \neq 0\}$
- $OutSplus_\sigma = \{a_i : (\exists j)\sigma(2j) = i+1\} \cup \{a_i : \sigma(2i+1) = 0\}$

We define $\mathcal{T}^{\mathcal{F}}$ as the set of σ so that

1. *In_σ is conflict-free;*
2. *$In_\sigma \cap Out_\sigma = \emptyset$;*
3. *$InSplus_\sigma \cap OutSplus_\sigma = \emptyset$;*
4. *If $\sigma(2j) = k+1$, then $a_k \rightarrowtail a_j$;*
5. *If $\sigma(2j+1) = k+1$, then $a_k \rightarrowtail a_j$;*
6. *For $j, k < |\sigma|$, if $a_k \in OutSplus_\sigma$ and $a_j \in In_\sigma$ then $a_j \not\rightarrowtail a_k$;*
7. *For $n, m < |\sigma|$, if $a_n \in OutSplus_\sigma$ and $a_m \in In_\sigma$, then $a_n \not\rightarrowtail a_m$ (i.e., σ does not contradict $S \subseteq f_{\mathcal{F}}(S)$).*

Theorem 5.12. *The complete extensions of $\mathcal{F}$ are in bijection with the set of paths $[\mathcal{T}^{\mathcal{F}}]$.*

Proof. Let S be any complete extension. We can define π from S as described at the beginning of this section. It is straightforward to verify that each condition (1-7) of Definition 5.11 is satisfied by $\pi \restriction_n$ for each n.

Given a path $\pi \in [\mathcal{T}^{\mathcal{F}}]$, we can define sets $S = \{i : \pi(2i) = 0\}$ and $U = \{i : \pi(2i+1) \neq 0\}$. We note that when $\sigma \preceq \tau$, $In_\sigma \subseteq In_\tau$. It follows from this fact, as in the previous theorems, that $S = \bigcup_{\sigma \prec \pi} In_\sigma$. Similarly, $U = \bigcup_{\sigma \prec \pi} \mathrm{InSplus}_\sigma$.

Next we see that $U = S^+$. If $n \in U$, then $\pi(2n+1) \neq 0$ and by condition 5, we have $a_{\pi(2n+1)-1}$ attacks a_n. But then $a_{\pi(2n+1)-1} \in In_{\pi\restriction_{2n+2}}$, so $a_{\pi(2n+1)-1} \in S$. Thus

$U \subseteq S^+$. On the other hand if $n \notin U$, then condition 6 for all σ of length $> 2n+1$ ensures that there is no $a_j \in S$ so $a_j \rightarrowtail a_n$. Thus $S^+ \subseteq U$.

Finally, we verify that S is complete. S is clearly conflict free by Condition 1. Condition 7 ensures that any $a_m \in S$ is also in $f_{\mathcal{F}}(S)$, since if $n \notin U$, then $a_n \not\rightarrowtail a_m$. To see that $f_{\mathcal{F}}(S) \subseteq S$, note that any $a_n \notin S$ has $\pi(n) \neq 0$ and $a_{\pi(n)-1} \notin S^+$ and $a_{\pi(n)-1} \rightarrowtail a_n$ by condition 4. Thus $a_n \notin f_{\mathcal{F}}(S)$. □

Remark 5.13. The map from paths $\pi \in [\mathcal{T}^{\mathcal{F}}]$ to complete extensions $S \subseteq A_{\mathcal{F}}$ is computable, but to compute $\pi \in \mathcal{T}^{\mathcal{F}}$ we need both S and S^+. Thus, we are able to say that for any computable AF $\mathcal{F}_e$, the set of Turing degrees of pairs (S, S^+) where S is a complete extension is always Spec($\mathcal{T}$) for a computable tree $\mathcal{T}$, but note that S and S^+ are not generally of the same Turing degree. Thus, we are currently unable to fully characterize $\text{Spec}_{co}^{\neg\emptyset}$ or $\text{Spec}_{infco}^{\neg\emptyset}$.

Corollary 5.14. *$\text{Uni}_{infco}^{\infty}$ is Π_1^1.*

Proof. We can alter the tree $\mathcal{T}^{\mathcal{F}_e}$ as in Corollary 5.4 to get a new tree $\widehat{\mathcal{T}}$ so that the paths through $\widehat{\mathcal{T}}$ are in bijection with the infinite complete extensions. Then $e \in \text{Uni}_{infco}^{\infty}$ if and only if $\widehat{\mathcal{T}}$ has a unique path. Theorem 3.10 shows that this is a Π_1^1 condition. □

Corollary 5.15. *Fix $\sigma \in \{ad, stb, co, infad, infstb, infco\}$ and suppose that $\mathcal{F}_e$ has only countably many non-empty σ extensions. Then there is a hyperarithmetical set S so that $\{a_n : n \in S\}$ is a σ extension of $\mathcal{F}_e$.*

Proof. If $\sigma \in \{ad, stb, infad, infstb\}$, let $\mathcal{T}$ be a tree so that $\text{Spec}_{\sigma}^{\neg\emptyset}(\mathcal{F}_e) = \text{Spec}(\mathcal{T})$. If $\sigma \in \{co, infco\}$, let $\mathcal{T}$ be a tree so that the set of Turing degrees of pairs $(S, S+)$ of σ extensions is exactly Spec($\mathcal{T}$). Then $\mathcal{T}$ is a computable tree with countably many paths. By a classic result of Kreisel [21, Theorem 3.9], $\mathcal{T}$ has a hyperarithmetical path, which corresponds to a hyperarithmetical S so that $\{a_n : n \in S\}$ is a σ extension of $\mathcal{F}_e$. □

6. Conclusion

In this paper, we initiated a systematic exploration of the complexity issues inherent to infinite argumentation frameworks. To pursue this direction, we employed computability-theoretic techniques which are ideally suited for assessing the complexity of infinite mathematical objects. Our focus was on the credulous and skeptical acceptance of arguments, as well as the existence and uniqueness of extensions, for admissible, complete, and stable semantics. We introduced and explored new semantics that are meaningful exclusively in the infinite setting, concerning the existence of infinite extensions that satisfy a given semantics σ. The computational problems we examined were found to be maximally complex, properly belonging to the complexity classes of Σ_1^1 and Π_1^1 sets.

A plethora of intriguing questions regarding the complexity of infinite AFs remains open. First, we shall fill the gaps that we left behind (such as proving the Π_1^1-hardness for $\text{Skept}_{co}^{\infty}$). Next, we aim at investigating whether the computational problems considered in this paper become more tractable if we restrict to special classes of AFs, such as the *finitary* ones (i.e., those in which each argument receives finitely many attacks only [7]). Finally, future research will extend our analysis to analogous problems associated with other key semantics for AFs, including grounded, preferred, and ideal semantics. Given that the definitions of these semantics are more intricate than those we examined here, we anticipate the need for additional techniques to thoroughly analyze them.

Acknowledgments

Andrews was partially supported by NSF grant DMS-2348792. San Mauro is a member of INDAM-GNSAGA.

References

[1] P. Baroni, F. Cerutti, P. E. Dunne, M. Giacomin, Automata for infinite argumentation structures, Artificial Intelligence 203 (2013) 104–150.

[2] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artificial intelligence 77 (1995) 321–357.

[3] A. J. García, G. R. Simari, Defeasible logic programming: An argumentative approach, Theory and practice of logic programming 4 (2004) 95–138.

[4] P. Baroni, M. Giacomin, G. Guida, Self-stabilizing defeat status computation: dealing with conflict management in multi-agent systems, Artificial Intelligence 165 (2005) 187–259.

[5] B. Verheij, Deflog: on the logical interpretation of prima facie justified assumptions, Journal of Logic and Computation 13 (2003) 319–346.

[6] M. Caminada, N. Oren, Grounded semantics and infinitary argumentation frameworks, in: Proceedings of the 26th Benelux Conference on Artificial Intelligence, BNAIC, 2014, pp. 25–32.

[7] R. Baumann, C. Spanring, Infinite argumentation frameworks: On the existence and uniqueness of extensions, in: Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation: Essays Dedicated to Gerhard Brewka on the Occasion of his 60th Birthday, Springer, 2015, pp. 281–295.

[8] P. Baroni, F. Cerutti, P. E. Dunne, M. Giacomin, Computing with infinite argumentation frameworks: The case of afras, in: Theorie and Applications of Formal Argumentation: First International Workshop, TAFA 2011. Barcelona, Spain, July 16-17, 2011, Springer, 2012, pp. 197–214.
[9] S. Bistarelli, F. Santini, et al., Weighted argumentation., FLAP 8 (2021) 1589–1622.
[10] P. E. Dunne, Coherence in finite argument systems, Artificial Intelligence 141 (2002) 187–203.
[11] P. E. Dunne, The computational complexity of ideal semantics, Artificial Intelligence 173 (2009) 1559–1591.
[12] W. Dvořák, S. Woltran, Complexity of semi-stable and stage semantics in argumentation frameworks, Information Processing Letters 110 (2010) 425–430.
[13] V. Brattka, P. Hertling, Handbook of computability and complexity in analysis, Springer, 2021.
[14] K. V. Velupillai, Computable foundations for economics, Routledge, 2012.
[15] G. Jäger, J. Rogers, Formal language theory: refining the chomsky hierarchy, Philosophical Transactions of the Royal Society B: Biological Sciences 367 (2012) 1956–1970.
[16] P. Baroni, M. Giacomin, Semantics of abstract argument systems, Argumentation in artificial intelligence (2009) 25–44.
[17] P. E. Dunne, M. Wooldridge, Complexity of abstract argumentation, Argumentation in artificial intelligence (2009) 85–104.
[18] H. Rogers Jr, Theory of recursive functions and effective computability, MIT press, 1987.
[19] S. C. Kleene, Arithmetical predicates and function quantifiers, Transactions of the American Mathematical Society 79 (1955) 312–340.
[20] A. Kechris, Classical descriptive set theory, volume 156, Springer Science & Business Media, 2012.
[21] D. Cenzer, Π^0_1 Classes in Computability Theory, in: Studies in Logic and the Foundations of Mathematics, volume 140, Elsevier, 1999, pp. 37–85.