Computing with Infinite Argumentation Frameworks: the Case of AFRAs

Pietro Baroni[1], Federico Cerutti[1], Paul E. Dunne[2], Massimiliano Giacomin[1]

[1] Dipartimento di Ingegneria dell'Informazione, University of Brescia, via Branze, 38, 25123, Brescia, Italy
{pietro.baroni,federico.cerutti,massimiliano.giacomin}@ing.unibs.it
[2] Department of Computer Science, Ashton Building, University of Liverpool, Liverpool, L69 7ZF, United Kingdom
P.E.Dunne@liverpool.ac.uk

Abstract. In recent years a large corpus of studies has arisen from Dung's seminal abstract model of argumentation, including several extensions aimed at increasing its expressiveness. Most of these works focus on the case of finite argumentation frameworks, leaving the potential practical applications of infinite frameworks largely unexplored. In the context of a recently proposed extension of Dung's framework called AFRA (Argumentation Framework with Recursive Attacks), this paper makes a first step to fill this gap. It is shown that, under some reasonable restrictions, infinite frameworks admit a compact finite specification and that, on this basis, computational problems which are tractable for finite frameworks may preserve the same property in the infinite case. In particular we provide a polynomial-time algorithm to compute the finite representation of the (possibly infinite) grounded extension of an AFRA with infinite attacks. An example concerning the representation of a moral dilemma is introduced to illustrate and instantiate the proposal and gives a preliminary idea of its potential applicability.

1 Introduction

Infinite argumentation frameworks, though encompassed by Dung's theory of abstract argumentation [6], have received relatively limited attention in the literature so that their use as a modelling tool and the relevant computational issues are largely unexplored.

This paper provides a first step towards filling this gap, by considering the case of existence of infinite attacks in a recently proposed extension of Dung's framework called AFRA (*Argumentation Framework with Recursive Attacks*) [2] where "attacks" may themselves be attacked by arguments. The idea of encompassing attacks to attacks in abstract argumentation framework has been first considered in [4], and subsequently investigated and developed, for instance, in [2, 11, 13]. Computational issues in this kind of extended frameworks have been first addressed in [8] for the finite case of EAF [13]. In this paper, we show that, under some mild restrictions, an AFRA with infinite attacks can be represented through a deterministic finite automaton (DFA), which provides the basis for the efficient solution of semantics-related computational problems. To demonstrate this, we show in particular that a DFA representing the (possibly infinite) grounded extension of an AFRA with infinite attacks can be derived in polynomial time from the DFA representing the AFRA itself.

From a general perspective, the ultimate aim of this paper is to provide an enabling technique for practical applications of infinite argumentation frameworks. While this is a largely open issue, we illustrate the theoretical concepts developed throughout the paper using a preliminary example concerning moral dilemma representation. Of course, the value of the methodology goes beyond both the simple example at hand and the use of the AFRA framework. Indeed the main contribution of this paper is twofold: on one hand, we address the topic of representing an Argumentation Framework through a formal language; and, secondly, we show that this kind of representation can be useful to compute semantics extensions also in the case of infinite Argumentation Frameworks.

The paper is organized as follows. After recalling the preliminary background concepts in Sect. 2, we provide an example encompassing infinite attacks in Sect. 3 and discuss specification mechanisms for AFRAs with infinite attacks in Sect. 4. Section 5 describes the actual specification mechanism adopted in the paper, called DFA$^+$, and Sect. 6 provides a polynomial time algorithm to compute the (representation of) the (possibly infinite) grounded extension of an AFRA starting from its DFA$^+$ specification. Finally Sect. 7 concludes the paper.

2 Preliminary Background

In this section we define the abstract argumentation models which are the core focus of this article: the AF model [6] with a finite set of arguments and the AFRA model [2].

Definition 1 *A* finite argumentation framework *(*AF*) is a pair* $\langle \mathcal{X}, \mathcal{A} \rangle$*, in which* $\mathcal{X}$ *is a finite set of* arguments *and* $\mathcal{A} \subseteq \mathcal{X} \times \mathcal{X}$ *is the* attack relationship*. A pair* $\langle x, y \rangle \in \mathcal{A}$ *is referred to as '*y is attacked by x*' or '*x attacks y*';* $x \in \mathcal{X}$ *is* acceptable with respect to $S \subseteq \mathcal{X}$ *if for every* $y \in \mathcal{X}$ *that attacks* x *there is some* $z \in S$ *that attacks* y*. The* characteristic function*,* $\mathcal{F} : 2^{\mathcal{X}} \rightarrow 2^{\mathcal{X}}$ *is the mapping which, given* $S \subseteq \mathcal{X}$*, returns the set of* $y \in \mathcal{X}$ *for which* y *is acceptable w.r.t.* S*. For any set* S *we define* $\mathcal{F}^0(S) = \emptyset$ *and for* $k \geq 1$ $\mathcal{F}^k(S) = \mathcal{F}(\mathcal{F}^{k-1}(S))$*. The* grounded *extension is the (unique) least fixed point of* $\mathcal{F}$*. We denote by* $GE(\langle \mathcal{X}, \mathcal{A} \rangle) \subseteq \mathcal{X}$ *the grounded extension of* $\langle \mathcal{X}, \mathcal{A} \rangle$*.*

Definition 2 *An* Argumentation Framework with Recursive Attacks *(*AFRA*) is described by a pair* $\langle \mathcal{X}, \mathcal{R} \rangle$ *where* $\mathcal{X}$ *is a (finite) set of arguments and* $\mathcal{R}$ *consists of pairs of the form* $\langle x, \alpha \rangle$ *where* $x \in \mathcal{X}$ *and* $\alpha \in \mathcal{X} \cup \mathcal{R}$*. For* $\alpha = \langle x, \beta \rangle \in \mathcal{R}$*, the* source *(src) and* target *(trg) of* α *are defined by* $src(\alpha) = x$ *and* $trg(\alpha) = \beta$*. In order to avoid a surfeit of brackets, we describe elements of* $\mathcal{R}$ *as finite length* sequences *of arguments, so that* $x_k \; x_{k-1} \; x_{k-2} \; \cdots \; x_2 \; x_1 \in \mathcal{R}$ *if* $\{x_1, \ldots, x_k\} \subseteq \mathcal{X}$ *(note that an argument may occur* more than once *in this* *sequence),* $\langle x_2, x_1 \rangle \in \mathcal{R}$ *(i.e.* $x_2 x_1 \in \mathcal{R}$*) and* $\langle x_j \langle x_{j-1} \langle \cdots x_1 \rangle \rangle \rangle \in \mathcal{R}$*, with* $2 < j \leq k$*. Letting* $\mathcal{C} = \mathcal{R} \cup \mathcal{X}$*, for* $\alpha \in \mathcal{R}$ *and* $\beta \in \mathcal{C}$*,* α *is said to* defeat β *(*$\alpha \to \beta$*) whenever any of the following hold:*

1. $trg(\alpha) = \beta$
2. $trg(\alpha) = src(\beta)$ *i.e.* $\beta \in \mathcal{R}$*,* $\alpha = xy$ *and* $\beta = y\gamma$ *(*$y \in \mathcal{X}$*).*

Definition 3 *Let* $\langle \mathcal{X}, \mathcal{R} \rangle$ *be an* AFRA*,* $\alpha, \beta \in \mathcal{R}$*,* $\mathcal{V}, \mathcal{W} \in \mathcal{X} \cup \mathcal{R}$*,* $\mathcal{S} \subseteq \mathcal{X} \cup \mathcal{R}$*; then:*

- $\mathcal{W}$ *is acceptable w.r.t.* $\mathcal{S}$ *(or, equivalently* is defended *by* $\mathcal{S}$*) iff* $\forall \alpha \in \mathcal{R}$ *s.t.* $\alpha \to \mathcal{W}$ $\exists \beta \in \mathcal{S}$ *s.t.* $\beta \to \alpha$*;*
- *the characteristic function* $\mathbb{F}_{\langle \mathcal{X}, \mathcal{R} \rangle}$ *is defined as follows:* $\mathbb{F}_{\langle \mathcal{X}, \mathcal{R} \rangle} : 2^{\mathcal{X} \cup \mathcal{R}} \mapsto 2^{\mathcal{X} \cup \mathcal{R}}$*;* $\mathbb{F}_{\langle \mathcal{X}, \mathcal{R} \rangle}(\mathcal{S}) = \{\mathcal{V} | \mathcal{V} \textit{is acceptable w.r.t. } \mathcal{S}\}$*;*
- *the grounded extension (denoted as* $GE^{\text{AFRA}}(\langle \mathcal{X}, \mathcal{R} \rangle)$*) is the least fixed point of* $\mathbb{F}_{\langle \mathcal{X}, \mathcal{R} \rangle}$*.*

By considering the (Dung) style AF, $\langle \tilde{\mathcal{X}}, \tilde{\mathcal{R}} \rangle$ constructed from an AFRA $\langle \mathcal{X}, \mathcal{R} \rangle$ by $\tilde{\mathcal{X}} = \mathcal{X} \cup \mathcal{R}$ and $\tilde{\mathcal{R}} = \{\langle \alpha, \beta \rangle \ : \ \alpha \to \beta\}$ (for further details see [2]), a correspondence between semantics structures (e.g. the basic notions of conflict-free and admissible sets and the extensions of various semantics) in an AFRA $\langle \mathcal{X}, \mathcal{R} \rangle$ and the analogous (Dung style) structures within $\langle \tilde{\mathcal{X}}, \tilde{\mathcal{R}} \rangle$ is obtained. In particular we will exploit the fact that the grounded extension of an AFRA coincides with the (Dung style) grounded extension of the corresponding AF, i.e. with $GE(\langle \tilde{\mathcal{X}}, \tilde{\mathcal{R}} \rangle)$.

3 An Example: Moral Dilemmas

The recursive form of $\mathcal{R}$ in an AFRA, $\langle \mathcal{X}, \mathcal{R} \rangle$, in principle, admits the capability of describing infinite attack structures even though $\mathcal{X}$ is a finite set. To exemplify the potential utility of this kind of structures as a modelling tool we consider a case of moral dilemma.

Fred is the network administrator of a large company and among his duties he has to release emails, addressed to staff members, that have been accidentally blocked by the security filters. One day he gets a helpdesk request from Eve, a staff member and his best friend's wife, requesting the release of an email. As part of the procedure he has to ensure that the email is safe by scanning its contents. He finds out that it's actually an email addressed to Eve from her lover. He releases the email, and his initial reaction is to call his friend up and tell him about the affair. However, the law forbids him to reveal the information. This is a case of conflict of obligations, and, following [5, 1], we can model this situation with abstract argumentation[3].

[3] A detailed comparison of alternative argumentation-based approaches to practical reasoning is beyond the scope of this paper. The interested reader may refer to [2] for a comparison between AFRA and Modgil's EAF, or to [1] for an illustration of the modelling approach adopted in the example.

First of all, the reasons for the alternative actions can be represented as practical arguments [14]. Indeed, since Fred wants to be a good friend, then he should tell his friend what he knows (**T**), but since Fred wants to be a good citizen, then he should not (**D**). These two arguments are obviously attacking each other. Moreover, both **D** and **T** are related to values [5], respectively Legality and Friendship. These values can be represented as arguments (**L** and **F**) [14, 1] which affect the evaluation of the two practical arguments. For instance, in the case at hand, the value of Friendship would resolve the dilemma by making **T** prevail over **D**. From an argumentation point of view, this means that **F** would allow **T** to defeat **D**. This can be modelled by making ineffective the attack from **D** to **T** (α in Fig. 1) by attacking it through an attack (β) whose source is the value of Friendship **F**. This can be read as: even if the attack from **D** to **T** (α) holds because **D** and **T** support conflicting actions, nevertheless, in the case at hand, α is undermined by the moral commitment of **F**. Obviously, **L** states a similar moral commitment, namely making **D** prevail over **T**. This requires **L** to undermine both the attack (γ) from **T** to **D** and the attack (β) from **F** against α. In turn **F** should make ineffective the latter attacks (δ and η) and **L** and **F** will continue attacking each other's attacks forever. This infinite construction reveals an unresolved dilemma.

Finally, let us suppose that Fred chooses to pursue Legality rather than Friendship. This can be represented by another argument (**M**) (a "must" argument in the terminology of [1]). The argument **M** represents a choice between values in the case at hand. Therefore, **M** will undermine any moral commitment of **F** over the two actions **T** and **D** by attacking the (infinite number of) attacks whose source is **F**.

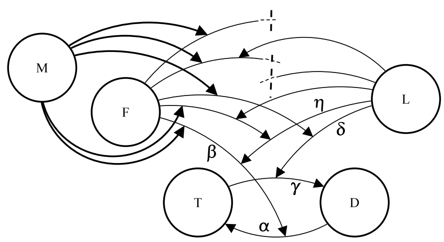

Fig. 1. Fred's dilemma

An AFRA representing Fred's dilemma is shown in Fig. 1. It consists of a finite set of arguments $\mathcal{X}_F = \{\mathbf{D}, \mathbf{T}, \mathbf{L}, \mathbf{F}, \mathbf{M}\}$ and of an infinite set of attacks $\mathcal{R}_F = \{\mathbf{DT}, \mathbf{TD}, \mathbf{LTD}, \mathbf{FDT}, \mathbf{FLTD}, \mathbf{LFDT}, \mathbf{LFLTD}, \mathbf{FLFDT}, \ldots, \mathbf{MFDT}, \mathbf{MFLTD}, \mathbf{MFLFDT}, \ldots\}$.

4 Representing $\mathcal{R}$ in AFRAs

Given the potential practical interest in AFRAs with infinite attacks, the following question arises.

> When $\mathcal{R}$ is infinite what characterises suitable specification mechanisms for describing $\mathcal{R}$?

In order to pursue this question, we need some terminology.

Definition 4 *For $\mathcal{X}$ a finite set of arguments, we denote by $\mathcal{X}^*$ the set of all finite length* sequences *(or* words*) that can be formed using arguments in $\mathcal{X}$ (noting this includes ε the so-called* empty sequence *comprising no arguments). Given $w \in \mathcal{X}^*$, $|w|$ denotes its* length, *i.e. the number of arguments occurring in its definition. Note that* repetitions *of the same arguments contribute to $|w|$ so that, e.g. $|x_1x_2x_1| = 3$ (and not 2). Given $w \in \mathcal{X}^*$ we will denote as $\bar{w}$ the sequence obtained by reversing the order of the symbols in w, namely, given $w = x_1x_2 \ldots x_n$, $\bar{w} = x_n \ldots x_2x_1$.*

Given $u = u_1u_2 \ldots u_r$ and $v = v_1v_2 \ldots v_k \in \Sigma^*$ we denote by $u \cdot v$ (or simply uv) the word w of length $k + r$ defined by $u_1u_2 \ldots u_rv_1v_2 \ldots v_k$. We note that $w \cdot \varepsilon = \varepsilon \cdot w = w$. We say that $\mathcal{L} \subseteq \mathcal{X}^*$ is an attack language over $\mathcal{X}$ *if $\mathcal{L}$ satisfies $\forall\ w \in \mathcal{L}\ w = xu$ with $x \in \mathcal{X}$ and either $|u| = 1$ or $u \in \mathcal{L}$.*

If $\mathcal{L}$ is an attack language over $\mathcal{X}$, then the pair $\langle \mathcal{X}, \mathcal{L} \rangle$ certainly describes an AFRA. Classical formal language and computability theory, see e.g. [12], provides a means of capturing the vague concept of "specification mechanism" via *Formal Grammars* and their associated machine models. As well known, given a set of symbols Σ a *formal grammar* G specifies the derivation of a language $L(G) \subseteq \Sigma^*$ called *language generated by G*. A language, $L \subseteq \Sigma^*$, is *recognisable* if there is a formal grammar G for which $w \in L$ if and only if $w \in L(G)$.

As a starting point for "specification mechanisms" for attack languages we can consider descriptions which are formal grammars (so that $\Sigma = \mathcal{X}$ in such cases).

Unsurprisingly, arbitrary attack languages have unhelpful computational properties.

Proposition 1 *Let $\mathcal{X} = \{x, y\}$. There are attack languages, L, over $\mathcal{X}$ which are not recognisable, i.e. for which there is no formal grammar G for which $L(G) = L$.*

Proof. In view of the correspondence from the fact that $L \subseteq \Sigma^*$ is *recursively enumerable* if and only if there is an unrestricted grammar, G such that $L(G) = L$, it suffices to show that there are attack languages which fail to be r.e. First recall that any TM program, M, can be associated with a finite length *codeword*, $\beta(M)$, (over the alphabet $\{0, 1\}$) in such a way that given $\beta(M)$ the behaviour of M can be reproduced by another TM program. Furthermore, the language corresponding to the set of valid encodings, i.e. $CODE = \{w \in$

$\{0,1\}^* : w = \beta(M)$ for some TM program, $M\}$ is recursive.[4] With such encodings it is known that the language $L^{\varepsilon}_{\neg HALT} \subset \{0,1\}^*$ given by $\{\beta(M)$: The TM program, M, **fails to halt** given the empty word as input$\}$ is not r.e.

Now since $CODE \subset \{0,1\}^*$ we can *order* the set of all TM programs simply by ordering words[5] within $\{0,1\}^*$, so that the "first" TM program (M_1) is the first word, w_1 in this ordering of $\{0,1\}^*$ for which $w_1 \in CODE$, the "second" program (M_2) the second word, w_2 in the ordering for which $w_2 \in CODE$, and so on.

We are now ready to define a suitable attack language, $\mathcal{R} \subset \{x,y\}^*$ establishing the proposition's claim: $\mathcal{R} = \{ xy^k : k \geq 2 \text{ and } M_k \in L^{\varepsilon}_{\neg HALT} \} \cup \{y^n : n \geq 2\}$. This is easily seen to be an attack language[6] and, furthermore, cannot be r.e. For suppose, $\mathcal{R}$ is r.e. with AL a TM accepting exactly the words in $\mathcal{R}$ then $L^{\varepsilon}_{\neg HALT}$ could be shown r.e. as follows: given $\beta(M)$ determine the index k for which M is the k'th TM program. Then $\beta(M) \in L^{\varepsilon}_{\neg HALT}$ if and only if xy^k is accepted by AL. □

As a consequence of Propn. 1 there will be attack languages for which it is not possible to present any specification (as a formal grammar). Of course the nature of such languages is unlikely to be of practical concern: Propn. 1 merely establishes a technical limitation affecting attack languages but certainly does not invalidate their use. In practice we would wish to consider only attack languages that are presented in some "verifiable form". What is the notion of "verifiable form" intended to capture here? Presented with a formal grammar G, there are two immediate issues which we would like to ensure can be addressed:

Q1. How easily can it be verified that $L(G)$ *does* describe an attack language?
Q2. Assuming $L(G)$ is verified as describing *some* attack language, $\mathcal{R}$ over $\mathcal{X}$, given $\alpha \in \mathcal{X}^*$ how easily can it be decided whether α *is* an attack in $\langle \mathcal{X}, \mathcal{R} \rangle$, i.e. whether $\alpha \in L(G)$?

It can be easily derived from Rice's Theorem (see, e.g. [12, pp. 185–195]) that unrestricted grammars face problems with respect to Q1.

Proposition 2 *Given an unrestricted grammar G, the problem of determining if $L(G)$ is an attack language is undecidable.*

On the other hand the family of *regular languages* [12] provides the basis for a positive result, using automata as representation mechanism.

Definition 5 *A* deterministic finite automaton *(*DFA*) is defined via a 5-tuple, $M = \langle \Sigma, Q, q_0, F, \delta \rangle$ where $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$ is a finite set of input symbols,*

[4] See e.g. [7, Ch. 4] or any standard introductory text on computability, such as [12, Ch. 8.3].

[5] For example using the standard lexicographic ordering under which $0 <_{\text{lex}} 1$ and $u <_{\text{lex}} w$ whenever $|u| < |w|$.

[6] The reader concerned by the fact that this includes a self-attacking argument (y) may note that we may use $xy^k x$ and $y^n x$ ($n \geq 1$) to achieve the same effect without self-attacking arguments.

$Q = \{q_0, q_1, \ldots, q_m\}$ *a finite set of* states*;* $q_0 \in Q$ *the* initial *state;* $F \subseteq Q$ *the set of* accepting *states; and* $\delta : Q \times \Sigma \to Q$ *the* state transition *function. For* $q \in Q$ *and* $w \in \Sigma^*$*, the* reachable state *from* q *on input* w *is*

$$\rho(q, w) \quad = \quad \begin{cases} q \text{ if } w = \varepsilon \\ \delta(q, w) \text{ if } |w| = 1 \\ \delta(\rho(q, u), x) \text{ if } w = u \cdot x \end{cases}$$

A sequence $w = w_1 w_2 \ldots w_n \in \Sigma^*$ *is accepted by the* DFA $\langle \Sigma, Q, q_0, F, \delta \rangle$ *if* $\rho(q_0, \bar{w}) = \rho(q_0, w_n w_{n-1} \ldots w_1) \in F$*, i.e. the sequence of states (consistent with the state transition function* δ*) which processes every symbol in* w in reverse order *ends in an accepting state. For a* DFA*,* $M = \langle \Sigma, Q, q_0, F, \delta \rangle$*,* $L(M)$ *is the subset of* Σ^* *accepted by* M*.*

Fact 1 *The language* $L \subseteq \Sigma$ *is regular if and only if there is a* DFA $M = \langle \Sigma, Q, q_0, F, \delta \rangle$ *for which* $L(M) = L$*.*

The following lemma shows that the conditions for an automaton to recognize an attack language are relatively simple.

Lemma 1. *Let* $M = \langle \mathcal{X}, Q, q_0, F, \delta \rangle$ *be a* DFA*. Then* $L(M)$ *is an attack language if and only if both the following conditions hold:*

C1. $\forall\, w \in \{\varepsilon\} \cup \mathcal{X}$, $\rho(q_0, w) \notin F$.
C2. $\forall\, q \in (Q \setminus \{q_0\})$, $\forall x \in \mathcal{X}$ *if* $q' = \delta(q, x) \notin F$ *then* $\forall w \in \mathcal{X}^*$ *it holds that* $\rho(q', \bar{w}) \notin F$.

Proof. Suppose first that $L(M)$ is an attack language. Since every $w \in L(M)$ satisfies $|w| \geq 2$ it is immediate that M satisfies C1. To see that C2 must hold, consider any $q \in (Q \setminus \{q_0\})$ and $x \in \mathcal{X}$ such that $q' = \delta(q, x) \notin F$. Furthermore consider any $u \in \mathcal{X}^*$ such that $q = \rho(q_0, \bar{u})$. Since $q \neq q_0$, $|u| \geq 1$ and, since $q' = \delta(q, x) \notin F$, $xu \notin L(M)$ and $|xu| \geq 2$. Since $L(M)$ is an attack language $\nexists p \in \mathcal{X}^*$ such that $p = vxu \in L(M)$, i.e. it is not possible to reach an accepting state from $q' = \delta(q, x)$.

For the converse direction, we show that if M satisfies both C1 and C2 then $L(M)$ is an attack language, i.e. $\forall\, w = xu \in L(M)$ either $|u| = 1$ or $u \in L(M)$. Since C1 holds, it is immediate that $|w| \geq 2$ for every $w \in L(M)$. Suppose now $w = yu \in L(M)$ with $|u| > 1$. Assume by contradiction $u \notin L(M)$, i.e. letting $q' = \rho(q_0, \bar{u})$ it holds that $q' \notin F$. Since $|u| > 1$ it must be the case that $q' = \delta(q, x)$ for some x with $q \neq q_0$. By C2, this implies that $\forall w \in \mathcal{X}^*$ $\rho(q', \bar{w}) \notin F$ which contradicts $w = yu \in L(M)$, as this would entail $\delta(q', y) \in F$. □

The desired result in Theorem 1 follows directly from Fact 1 and Lemma 1.

Theorem 1 *Let* $M = \langle \mathcal{X}, Q, q_0, F, \delta \rangle$ *be a* DFA *defining the regular* language*,* $L(M) \subseteq \mathcal{X}^*$*. The problem of verifying that* $L(M)$ *is an attack language is polynomial time decidable.*

Proof. Given a DFA, $M = \langle \mathcal{X}, Q, q_0, \delta, F\rangle$ from Lemma 1, in order to verify that $L(M)$ is an attack language it suffices to confirm that M satisfies the conditions C1 and C2 and that these can tested in time polynomial in $|Q|$.

To check that C1 holds we need only confirm that $q_0 \notin F$ giving $\varepsilon \notin L(M)$ and for each $x \in \mathcal{X}$ that $\delta(q_0, x) \notin F$, so that $x \notin L(M)$. To test condition C2, for all non-accepting states q' for which there exists a transition from a state different from q_0, it has to be verified that for all $w \in \mathcal{X}^*$ $\rho(q', \bar{w}) \notin F$. This, however, is simply a directed path problem, i.e. verifying that there is no path from q' to any state in F which is easily solved in polynomial time, e.g. by carrying out a breadth-first search of states reachable from q'. □

Finally, as to question Q2, given M a DFA describing the attack language $L(M)$ and $w \in \mathcal{X}^*$, we can decide if w is an an attack in the AFRA $\langle \mathcal{X}, L(M)\rangle$ in polynomial time simply by confirming that $\rho(q_0, \bar{w}) \in F$.

5 The DFA$^+$ Representation of AFRAs

Expressing $\mathcal{R}$ within an AFRA, $\langle \mathcal{X}, \mathcal{R}\rangle$ via a DFA, M for which $L(M) = \mathcal{R}$ turns out to have some useful computational benefits in addition to verifiability and deciding whether a specified attack is present. We will demonstrate these advantages as far as the problem of computing the grounded extension is concerned. To this purpose we have first to introduce a representation of a whole AFRA (not just the attack relation) as an automaton and analyze its properties. Given an AFRA $\langle \mathcal{X}, \mathcal{R}\rangle$ where $\mathcal{R} \subset \mathcal{X}^*$ is a regular language represented as a DFA $\mathcal{M} = \langle \mathcal{X}, Q_{\mathcal{M}}, q_0, F_{\mathcal{M}}, \delta\rangle$, it is easy to obtain a representation of $\langle \mathcal{X}, \mathcal{R}\rangle$ as a single DFA $\mathcal{M}^+ = \langle \mathcal{X}, Q_{\mathcal{M}^+}, q_0, F_{\mathcal{M}^+}, \delta^+\rangle$ (indicated for the sake of brevity as DFA$^+$ in the following) such that for any $w \in \mathcal{X}^*$ it holds $w \in L(\mathcal{M}^+)$ if and only if $w \in \mathcal{X} \cup \mathcal{R}$. Let us notice that, in general, there are infinite DFA$^+$s representing a single AFRA. This may raise the problem of defining a canonical DFA$^+$ representation for each AFRA. This problem, not considered in the paper, is left for future work. In the following we will provide some general results that hold for any DFA$^+$ representing an AFRA.

Figure 2 shows $\mathcal{M}^+_F$, a DFA$^+$ which accepts all the words of the regular language $\mathcal{R}_F$ describing Fred's dilemma.

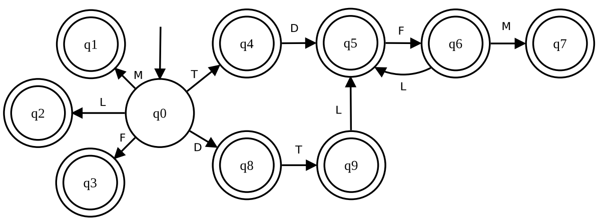

Fig. 2. A DFA$^+$ for Fred's dilemma (double circles represent accepting states)

We introduce also some handy notation concerning neighbor states and "input" symbols for a given state. For $p \in Q_{\mathcal{M}^+}$ we define $state - out(p) = \{ q \in Q_{\mathcal{M}^+} : \exists\, x \in \mathcal{X}$ for which $q = \delta^+(p, x)\}$. For instance, in $\mathcal{M}^+_F$, $state - out(q_0) = \{q_1, q_2, q_3, q_4, q_8\}$. For $p \in F_{\mathcal{M}^+}$ we define $sym - in(p) = \{x \in \mathcal{X} : \exists\, q \in Q_{\mathcal{M}^+}$ for which $p = \delta^+(q, x)\}$ and $state - in(p) = \{ q \in Q_{\mathcal{M}^+} : \exists\, x \in \mathcal{X}$ for which $p = \delta^+(q, x)\}$. In $\mathcal{M}^+_F$, $sym - in(q_5) = \{D, L\}$ and $state - in(q_5) = \{q_4, q_9, q_6\}$.

It is now useful to point out several properties of the DFA$^+$ representation (we will implicitly assume that each accepting state is reachable from q_0, as it should be in order to avoid useless parts in the automaton).

First we can partition the accepting states in $F_{\mathcal{M}^+}$ into two sets: *argument states* and *attack states*.

Argument states are in one-to-one correspondence with the elements of $\mathcal{X}$ and are reachable in one step from the initial state q_0: they represent the "additional part" of the DFA$^+$ w.r.t. the DFA representation. Formally $\forall x \in \mathcal{X}$ $\exists q \in F_{\mathcal{M}^+}$ such that $\delta^+(q_0, x) = q$ and $sym - in(q) = \{x\}$. For each $x \in \mathcal{X}$ we will denote the corresponding argument state as $argst(x)$ and, conversely, if $q = argst(x)$ we will say that $x = reparg(q)$. For the whole set of arguments $\mathcal{X}$ in a DFA$^+$ representation we define $ArgS(\mathcal{M}^+) \triangleq \{argst(x) \mid x \in \mathcal{X}\}$. Hence, $ArgS(\mathcal{M}^+_F) = \{q_1, q_2, q_3, q_4, q_8\}$.

In AFRA an argument can receive only direct defeats from other arguments: an argument x is defeated by an argument y if and only if $\langle x, y\rangle \in \mathcal{R}$ namely if the corresponding two-length string in $\mathcal{X}^*$ is accepted by the DFA$^+$ (and of course by the original DFA). Formally we can identify the set of direct defeaters of an argument x as $dirdef(x) \triangleq \{y \in \mathcal{X} \mid \delta^+(argst(x), y) \in F_{\mathcal{M}^+}\}$. Of course an argument x is *unattacked* in AFRA if and only if $dirdef(x) = \emptyset$. The set of unattacked arguments will be denoted as $unatt - args(\mathcal{M}^+)$. The above definitions can be extended from arguments to argument states in the obvious way.

Attack states are all the accepting states which are not argument states and are defined as $AttS(\mathcal{M}^+) \triangleq F_{\mathcal{M}^+} \setminus ArgS(\mathcal{M}^+)$. Hence, $AttS(\mathcal{M}^+_F) = \{q_5, q_6, q_7, q_9\}$. Every attack state q in a DFA$^+$ (and in the original DFA) corresponds to a (possibly infinite) subset of $\mathcal{R}$, namely to a (nonempty) set of elements of the corresponding attack language, denoted as $AttL(q)$. Formally, for any $q \in AttS(\mathcal{M}^+)$ $AttL(q) \triangleq \{r \in \mathcal{R} \mid \rho(q_0, \bar{r}) = q\}$. Given $r \in AttL(q)$ we will say that q is the *representative state* of r, denoted as $q = repst(r)$. Of course, $\forall r \in \mathcal{R}\ \exists! q \in AttS(\mathcal{M}^+) \mid q = repst(r)$.

An element r of $\mathcal{R}$ can have both direct defeaters and indirect defeaters (see 1. and 2. in Def. 2). A direct defeater is any argument x which is the source of an attack whose target is r, and then $xr \in \mathcal{R}$. It can then be observed that given an attack state q all elements of $AttL(q)$ have the same direct defeaters. Formally, for any $q \in AttS(\mathcal{M}^+)$ we define $dirdef(q) \triangleq \{x \in \mathcal{X} \mid \delta^+(q, x) \in F_{\mathcal{M}^+}\}$ and for any $r \in \mathcal{R}$ $dirdef(r) \triangleq dirdef(repst(r))$.

An indirect defeater is any argument x which is the source of an attack whose target is the source of r: $indirdef(r) \triangleq dirdef(src(r))$.

Given an attack state q it can be noted that the source of any attack represented by q corresponds to one of the elements of $sym-in(q)$: in fact any element of $sym-in(q)$ is the first symbol of some of the elements of the attack language accepted by q. By extension, we can hence define the indirect defeaters of any $q \in AttS(\mathcal{M}^+)$: $indirdef(q) \triangleq \bigcup_{r \in AttL(q)} indirdef(r) = \bigcup_{x \in sym-in(q)} dirdef(x)$.

The whole set of defeaters of an element r of $\mathcal{R}$ will be denoted as $totdef(r) \triangleq dirdef(r) \cup indirdef(r)$. Analogously, for a state q, $totdef(q) \triangleq dirdef(q) \cup indirdef(q)$. We say that an attack state q is unattacked if $totdef(q) = \emptyset$. For instance in Fig. 2 q_7 is unattacked while $totdef(q_5) = \{\mathbf{F}, \mathbf{T}\}$. In the following we will use the term unattacked states to refer collectively to both unattacked argument states and unattacked attack states. It can be noted that if an attack state q is unattacked then all elements of $AttL(q)$ are unattacked, but it does not hold that if $r \in \mathcal{R}$ is unattacked then $repst(r)$ is unattacked. In fact $totdef(r) = \emptyset$ implies $dirdef(repst(r)) = \emptyset$ but does not imply $indirdef(repst(r)) = \emptyset$ since $repst(r)$ might have indirect defeaters due to other elements of $AttL(q)$. On the other hand it can easily be observed that $totdef(r) = \emptyset$ implies also $indirdef(repst(r)) = \emptyset$ if $|sym - in(repst(r))| = 1$. Under this condition $r \in \mathcal{R}$ is unattacked if and only if $repst(r)$ is unattacked.

Since this is a desirable property, we need to introduce a transformation of DFA$^+$ aimed at ensuring the above condition while leaving unmodified the accepted language. This will be achieved by *splitting* some attack states of the DFA$^+$.

Definition 6 *An attack state p is* splittable *if $|sym - in(p)| > 1$. The set of splittable states of a* DFA$^+$ *$\mathcal{M}^+$ will be denoted as $split - states(\mathcal{M}^+)$.*

In $\mathcal{M}^+_F$, q_5 is splittable since $sym - in(q_5) = \{L, D\}$.

As explained above we need a *complete split* (*csplit* in the following) operator whose goal is transforming a DFA$^+$ (without affecting the language it accepts) so that in the resulting DFA$^+$ there are no splittable states. This is achieved by adding, for each splittable state p, a number $|sym - in(p)| - 1$ new accepting states. Accordingly a *split* operation w.r.t a splittable state can be defined as follows.

Definition 7 *For $\mathcal{M}^+ = \langle \mathcal{X}, Q_{\mathcal{M}^+}, q_0, F_{\mathcal{M}^+}, \delta^+ \rangle$ let p be a splittable state with $sym - in(p) = \{x_1, \ldots, x_n\}$, $(n > 1)$. The* DFA$^+$ *resulting by* splitting *p, $split(\mathcal{M}^+, p) = \langle \mathcal{X}, Q^{\mathrm{spl}}_{\mathcal{M}^+}, q_0, F^{\mathrm{spl}}_{\mathcal{M}^+}, \delta^{+\mathrm{spl}} \rangle$ is obtained by:*

S1. $Q^{\mathrm{spl}}_{\mathcal{M}^+} = Q_{\mathcal{M}^+} \cup \{p_2, \ldots, p_n\}$ *where $p_2, \ldots, p_n$ are new (accepting) states hence included also in $F^{\mathrm{spl}}_{\mathcal{M}^+}$.*
S2. Letting $p_1 = p$ the transition function $\delta^{+\mathrm{spl}}$ has, for $i = 1 \ldots n$:
$\delta^{\mathrm{spl}}(q', x_i) = p_i$ *if* $q' \in state - in(p) \wedge \delta(q', x_i) = p$
$\delta^{\mathrm{spl}}(p_i, y) = \delta(p, y)$
$\delta^{\mathrm{spl}}(q, y) = \delta(q, y)$ *if* $q \in Q_{\mathcal{M}^+} \setminus state - in(p)$

In words, a splittable state p is partitioned into several states p_i each with $sym-in(p_i) = \{x_i\}$ and the transitions from p to other states are replicated from each p_i to them. It can be observed that the application of the split operation:

- does not affect the language accepted by the DFA$^+$: for any splittable state p $L(\mathcal{M}^+) = L(split(\mathcal{M}^+, p))$;
- does not affect the cardinality of $sym - in(q)$ for any state $q \neq p$: in fact q may have additional incoming transitions from the elements p_i but they all correspond to elements already present in $sym - in(q)$;
- for each state p_i, letting x_i be the only element of $sym{-}in(p_i)$, in $split(\mathcal{M}^+, p)$ it holds that $dirdef(p_i) = dirdef(p)$ and $indirdef(p_i) = indirdef(x_i)$.

In virtue of the second point above, it can be noted that it is possible to extend the definition of the split operation to a set of splittable states: given a set P of splittable states of a DFA$^+$ $\mathcal{M}^+$, the result of the operation $split(\mathcal{M}^+, P)$ is the DFA$^+$ resulting from the application of $split(\mathcal{M}^+, p)$ for each $p \in P$ (the order of application of the operations $split(\mathcal{M}^+, p)$ does not matter).

Of course the *csplit* operation is obtained by applying the split operation to all splittable states of a DFA$^+$ $\mathcal{M}^+$: $csplit(\mathcal{M}^+) \triangleq split(\mathcal{M}^+, split{-}states(\mathcal{M}^+))$. It is easy to observe that the number of states of $split(\mathcal{M}^+, split - states(\mathcal{M}^+))$ is upper bounded by $|Q_{\mathcal{M}^+}| * |\mathcal{X}|$ hence the *csplit* operation can be carried out in polynomial time with respect to the number of states and arguments of $\mathcal{M}^+$.

Figure 3 depicts the result of the application of the *csplit* operator to $\mathcal{M}^+_F$. As we noticed before, q_5 is a splittable state (and it is the only one in $\mathcal{M}^+_F$). $csplit(\mathcal{M}^+_F)$ has an additional state w.r.t. $\mathcal{M}^+_F$, namely q'_5 with $sym - in(q'_5) = \{L\}$ while, after splitting, $sym - in(q_5) = \{D\}$. Moreover, as required by Def. 7, any outgoing transitions from the split state is replicated, giving rise to the transitions from q_5 to q_6 and from q'_5 to q_6, both triggered by F.

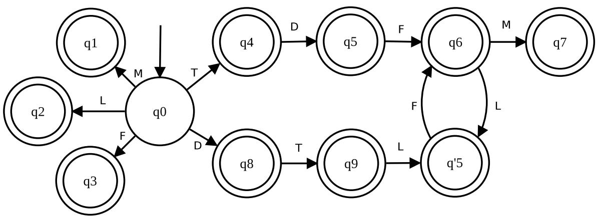

Fig. 3. Graphical representation of $csplit(\mathcal{M}^+_F)$

6 Computing the Grounded Extension with the DFA$^+$ Representation

In this section we show that the grounded extension of AFRAs with DFA$^+$ representation can be computed in polynomial time. Since the grounded extension of an AFRA includes both arguments and attacks, it may be infinite and therefore will, in turn, be expressed through a DFA$^+$, algorithmically derived from the one of the AFRA.

Before illustrating the algorithm we need to consider some properties of AFRAs and of the grounded extension.

First recall a characterization of the grounded extension for finitary argumentation frameworks [6].

Definition 8 *An argumentation framework $\langle \mathcal{X}, \mathcal{A} \rangle$ is* finitary *iff for each argument x there are only finitely many arguments in $\mathcal{X}$ which attack x.*

Proposition 3 *If an argumentation framework* AF *is finitary then $GE(\text{AF}) = \bigcup_{i=1\ldots\infty} \mathcal{F}^i(\emptyset)$ where $\mathcal{F}$ is the characteristic function of* AF *(Def. 1).*

It is now easy to see that, for any AFRA, the corresponding AF $\langle \tilde{\mathcal{X}}, \tilde{\mathcal{R}} \rangle$ (see Sect. 2) is finitary:

- the attackers of each element x of $\tilde{\mathcal{X}} \cap \mathcal{X}$ correspond to the direct defeaters of x in AFRA, which are at most $|\mathcal{X}|$;
- the attackers of each element r of $\tilde{\mathcal{X}} \cap \mathcal{R}$ correspond to the direct and indirect defeaters of r in AFRA, which are at most $2 * |\mathcal{X}|$.

On this basis we can now state some relatively straightforward conditions concerning the membership of AFRA arguments and attacks to $GE(\langle \tilde{\mathcal{X}}, \tilde{\mathcal{R}} \rangle) = GE^{\text{AFRA}}(\langle \mathcal{X}, \mathcal{R} \rangle)$, drawing relations between the characteristic function and defeaters in the DFA$^+$ representation.

Proposition 4 *Let $\langle \mathcal{X}, \mathcal{R} \rangle$ be an* AFRA *with* DFA$^+$ *representation and $\langle \tilde{\mathcal{X}}, \tilde{\mathcal{R}} \rangle$ be its corresponding* AF *with characteristic function $\mathcal{F}$, x be an element of $\tilde{\mathcal{X}} \cap \mathcal{X}$, r be an element of $\tilde{\mathcal{X}} \cap \mathcal{R}$. It holds that:*

1. $x \in \mathcal{F}^1(\emptyset)$ *iff* $dirdef(x) = \emptyset$
2. $r \in \mathcal{F}^1(\emptyset)$ *iff* $totdef(r) = \emptyset$
3. *for* $i \geq 2, x \in \mathcal{F}^i(\emptyset) \setminus \mathcal{F}^{i-1}(\emptyset)$ *iff* $\forall y \in dirdef(x)\ (totdef(yx) \cap \mathcal{F}^{i-1}(\emptyset)) \neq \emptyset \wedge \exists y \in dirdef(x) \mid (totdef(yx) \cap \mathcal{F}^{i-2}(\emptyset)) = \emptyset$
4. *for* $i \geq 2, r \in \mathcal{F}^i(\emptyset) \setminus \mathcal{F}^{i-1}(\emptyset)$ *iff* $\forall y \in totdef(r)\ (totdef(yr) \cap \mathcal{F}^{i-1}(\emptyset)) \neq \emptyset \wedge \exists y \in totdef(r) \mid (totdef(yr) \cap \mathcal{F}^{i-2}(\emptyset)) = \emptyset$

We can now introduce an algorithm (Alg. 1) which builds a DFA accepting the grounded extension of $\langle \mathcal{X}, \mathcal{R} \rangle$. The result of its execution on $\mathcal{M}_F^+$ is illustrated in Fig. 4. After splitting, in the first iteration of the **repeat** cycle the unattacked states q_1, q_2, q_3, q_7 are marked **in**(1) (note that q_7 has no indirect defeaters since q_1 is unattacked). Then, since $state - in(q_7) = \{q_6\}$, q_6 is marked **out** and removed from the set of accepting states. As a consequence, during the second iteration, q'_5 is unattacked and is marked **in**(2). Then, q_9 is marked **out** at line 9 of Alg. 1 and removed from the set of accepting states. Finally, in the third iteration, both q_5 and q_8 are unattacked (note in particular that q_5 is unattacked since $argst(\mathbf{D}) = q_8$ is unattacked). As a consequence they are marked **in**(3) and q_4 is marked **out** at line 9 of Alg. 1. The algorithm will then terminate at the following iteration.

Algorithm 1 Determining $GE(\langle \mathcal{X}, \mathcal{R} \rangle)$ in AFRAs

```
Input: DFA+ M+ = ⟨X, Q_M+, q0, F_M+, δ+⟩ with α ∈ L(M+) ⇔ α ∈ X ∪ R.
Output: DFA M_G = ⟨X, Q_G, q0, F_G, δ_G⟩ with α ∈ L(M_G) ⇔ α ∈ GE(⟨X̃, R̃⟩)
i := 0
M_i := csplit(M+); with M_i = ⟨X, Q_i, q0, F_i, δ_i⟩
repeat
   i := i + 1; M_i := M_{i−1};
   For each (unmarked) unattacked state q of M_i mark q as in(i).
   for each unattacked state q and every q′ ∈ state − in(q) ∩ F_i do
      Mark q′ as out and remove q′ from F_i.
   end for
   for each x ∈ X s.t. argst(x) is marked out do
      For each state q ∈ F_i with x ∈ sym − in(q) mark q as out and remove q from
      F_i.
   end for
until M_i = M_{i−1}
for any q ∈ F_i which is not marked in() do
   remove q from F_i
end for
return ⟨X, Q_i, q0, F_i, δ_i⟩
```

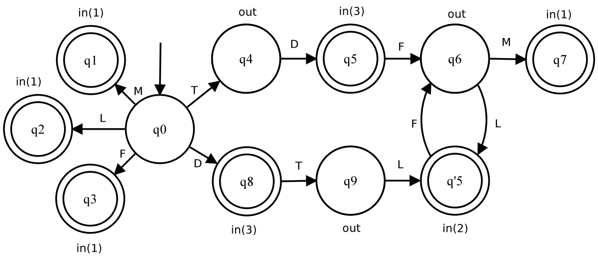

Fig. 4. The DFA$^+$ after the execution of Alg. 1 on $\mathcal{M}_F^+$

From an argumentation point of view, this result means that the arguments **M**, **L**, **F** and **D** are in the AFRA grounded extension, along with any attack whose source is one of **M**, **L**, and **D**. Therefore, the dilemma's solution is that Fred should not tell his friend what he knows, because in this situation the value of legality prevails over the value of friendship.

Turning back to technical results, correctness of Algorithm 1 follows from the following proposition.

Proposition 5 *Let $\mathcal{M}^+ = \langle \mathcal{X}, Q_{\mathcal{M}^+}, q_0, F_{\mathcal{M}^+}, \delta^+ \rangle$ with $\alpha \in L(\mathcal{M}^+) \Leftrightarrow \alpha \in \mathcal{X} \cup \mathcal{R}$ be a* DFA$^+$ *describing the* AFRA *$\langle \mathcal{X}, \mathcal{R} \rangle$, with corresponding* AF *$\langle \tilde{\mathcal{X}}, \tilde{\mathcal{R}} \rangle$, and $\mathcal{M}_i = \langle \mathcal{X}, Q_i, q_0, F_i, \delta_i \rangle$ the automaton produced by Algorithm 1 at the i-th* *iteration of the **repeat** cycle. For $i \geq 0$, let $T_i \subseteq F_i$ be the set of states*

$$T_i = \bigcup_{k=1}^{i} \{ q \in F_i : q \text{ is labelled } \textbf{in}(k) \text{ by Algorithm 1} \}$$

and $L_i = \{\alpha \in \mathcal{X}^* : \rho(q_0, \bar{\alpha}) \in T_i\}$. For every $i \geq 1$ $\alpha \in \mathcal{X} \cup \mathcal{R}$ is in L_i if and only if $\alpha \in \mathcal{F}_i(\emptyset)$, i.e. α is acceptable w.r.t. $\mathcal{F}_{i-1}(\emptyset)$ in $\langle \tilde{\mathcal{X}}, \tilde{\mathcal{R}} \rangle$.

Proof. The proof proceeds by induction on $i \geq 1$. For the inductive base we need $\alpha \in L_1$ if and only if $\alpha \in \mathcal{F}_1(\emptyset)$. Assume first that $\alpha \in L_1$: from l. 7 of Alg. 1 it follows that $q = \rho(q_0, \bar{\alpha})$ is unattacked after the *csplit* operation has been applied to $\mathcal{M}_0 = \mathcal{M}^+$. If q is an argument state (not affected by the *csplit* operation) it follows that $dirdef(\alpha) = \emptyset$ both in $\mathcal{M}_0 = \mathcal{M}^+$ and in $\mathcal{M}_1$: hence, by Prop. 4, $\alpha \in \mathcal{F}_1(\emptyset)$. If q is an attack state it either was unattacked in $\mathcal{M}_0 = \mathcal{M}^+$ or it became unattacked in $\mathcal{M}_1$ as a consequence of the splitting of a splittable state in $\mathcal{M}_0$. Taking into account the properties of the split operation discussed in Sec. 5, in both cases it holds that $\alpha \in AttL(q)$ and $totdef(\alpha) = \emptyset$: again by Prop. 4, $\alpha \in \mathcal{F}_1(\emptyset)$.

Assume now that $\alpha \in \mathcal{F}_1(\emptyset)$. From Prop. 4 one of the following two conditions holds: α is an argument with $dirdef(\alpha) = \emptyset$ or α is an attack with $totdef(\alpha) = \emptyset$. In the first case the state $q = argst(\alpha)$ is unattacked in $\mathcal{M}_0 = \mathcal{M}^+$ (and hence also in $\mathcal{M}_1$) and is marked as **in**(1) by l. 8, hence $q \in T_1$ and $\alpha \in L_1$. In the second case it follows that the state $q = repst(\alpha)$ is either unattacked or splittable in $\mathcal{M}_0 = \mathcal{M}^+$. In fact q can not have direct defeaters (since α has not), and either has not indirect defeaters (hence being unattacked) or has indirect defeaters (due to other elements of $AttL(q)$) hence being splittable. As a consequence, in both cases after the *csplit* operation on $\mathcal{M}_0$, in $\mathcal{M}_1$ $q = repst(\alpha)$ is unattacked and is marked as **in**(1) by l. 7, hence $q \in T_1$ and $\alpha \in L_1$.

Now inductively assume, for some $k \geq 1$, that for all $i \leq k$ $\alpha \in L_i$ if and only if $\alpha \in \mathcal{F}_i(\emptyset)$. We show $\alpha \in L_{k+1}$ if and only if $\alpha \in \mathcal{F}_{k+1}(\emptyset)$.

Consider any $\alpha \in L_{k+1}$: without loss of generality we may assume $\alpha \in L_{k+1} \setminus L_k$ (since $\mathcal{F}_k(\emptyset) \subseteq \mathcal{F}_{k+1}(\emptyset)$ and, via induction, we have $\alpha \in L_k$ if and only if $\alpha \in \mathcal{F}_k(\emptyset)$).

If α is an argument, namely $\alpha \in \tilde{\mathcal{X}} \cap \mathcal{X}$, it follows that $q = argst(\alpha) \in T_{k+1} \setminus T_k$. If α is an attack, namely $\alpha \in \tilde{\mathcal{X}} \cap \mathcal{R}$, it follows that $q = repst(\alpha) \in T_{k+1} \setminus T_k$.

In both cases, it holds that q is marked as **in**$(k+1)$ by l. 7, hence q is unattacked in $\mathcal{M}_{k+1}$ while it is not unattacked in $\mathcal{M}_k$. This means that any $p \in state-out(q)$ has already been marked as **out**. Moreover if α is an attack, also any argument state t such that $reparg(t) \in indirdef(\alpha)$ has already been marked as **out**.

The **out** marking can be carried out at l. 9 or l. 12 of Alg. 1. In the case of l. 9 p is marked as **out** since a state $q' \in state-out(p)$ has been marked as **in**(i) with $i \leq k$. This means that for any $\beta \in dirdef(\alpha)$ (with $\beta\alpha \in AttL(p)$ for some $p \in state-out(q)$ marked as **out** at l. 9) $\exists \gamma \in dirdef(\beta\alpha)$ such that $repst(\gamma\beta\alpha) = q'$ is marked as **in**(i) with $i \leq k$. By the inductive hypothesis, we have that a (direct) defeater γ of the attack $\beta\alpha$ is in $\mathcal{F}_k(\emptyset)$, hence α is defended by $\mathcal{F}_k(\emptyset)$ with respect to any $\beta \in dirdef(\alpha)$ (with $\beta\alpha \in AttL(p)$ for some $p \in state-out(q)$ marked as **out** at l. 9). With a similar reasoning, in the case α is an attack, we may also conclude that for any $\beta \in indirdef(\alpha)$ (with $\beta = reparg(t)$ for some argument state t marked as **out** at l. 9) $\exists \gamma \in dirdef(\beta)$ such that $repst(\gamma\beta) = q'$ is marked as $\mathbf{in}(i)$ with $i \le k$ and hence α is defended by $\mathcal{F}_k(\emptyset)$ with respect to any $\beta \in indirdef(\alpha)$ (with $\beta = reparg(t)$ for some argument state t marked as **out** at l. 9).

In the case of l. 12 p is marked out since $q' = argst(\beta)$ has been marked out with β the (only) element of $sym-in(p)$. It can be observed that any argument state can be marked as **out** only at l. 9 (to satisfy the condition for marking at l. 12 an argument state should be already marked as **out** according to l. 11). This means that $\exists q'' \in state-out(q')$ with q'' marked as $\mathbf{in}(i)$ with $i \le k$. By the inductive hypothesis $\exists \gamma \in dirdef(\beta)$ such that $\gamma\beta \in AttL(q'')$ and $\gamma\beta \in \mathcal{F}_k(\emptyset)$. This means that an (indirect) defeater of all elements of $AttL(p)$ belongs to $\mathcal{F}_k(\emptyset)$, hence α is defended by $\mathcal{F}_k(\emptyset)$ with respect to any attack in $AttL(p)$.

Summing up, it follows that $\mathcal{F}_k(\emptyset)$ defends α against any $\beta \in dirdef(\alpha)$ (either $\beta\alpha$ is attacked, case of l. 9, or β is attacked, case of l. 12) and, if α is an attack, $\mathcal{F}_k(\emptyset)$ defends α against any $\beta \in indirdef(\alpha)$ (β is attacked, case of l. 9). It ensues $\alpha \in \mathcal{F}_{k+1}(\emptyset)$.

Turning to the other side of the proof of the inductive step, assume now $\alpha \in \mathcal{F}_{k+1}(\emptyset)$. Again, without loss of generality, we may consider only the case $\alpha \in \mathcal{F}_{k+1}(\emptyset) \setminus \mathcal{F}_k(\emptyset)$.

If α is an argument, from case 3. of Prop. 4 it follows that $\forall \beta \in dirdef(\alpha)$ $(totdef(\beta\alpha) \cap \mathcal{F}^k(\emptyset)) \neq \emptyset \wedge \exists \beta \in dirdef(\alpha) \mid (totdef(\beta\alpha) \cap (\mathcal{F}^k(\emptyset) \setminus \mathcal{F}^{k-1}(\emptyset))) \neq \emptyset$. This implies that $\forall \beta \in dirdef(\alpha)$ α is defended by $\mathcal{F}^k(\emptyset)$ against β, namely there is an argument γ such that $\gamma\beta\alpha$ or $\gamma\beta$ belongs to $\mathcal{F}^k(\emptyset)$ (in both cases it must also hold $\gamma \in \mathcal{F}^k(\emptyset)$). Moreover, for one of these elements γ it must hold that either $\gamma\beta\alpha$ or $\gamma\beta$ belongs to $\mathcal{F}^k(\emptyset) \setminus \mathcal{F}^{k-1}(\emptyset)$.

By the inductive hypothesis, if follows that for any such γ, $argst(\gamma)$ is marked as $\mathbf{in}(i)$ with $i \le k$ and either $repst(\gamma\beta\alpha)$ or $repst(\gamma\beta)$ is marked as $\mathbf{in}(i)$ with $i \le k$ (again, for at least one of these elements, the mark is exactly $\mathbf{in}(k)$). It follows that $\forall \beta \in dirdef(\alpha)$ $repst(\beta\alpha)$ is marked out at an iteration $i \le k$ and one of these $repst(\beta\alpha)$ is marked out exactly at the iteration k. Hence $argst(\alpha)$ becomes unattacked, and hence is marked **in**, exactly at the iteration $k+1$ and $\alpha \in L_{k+1}$ as desired.

If α is an attack, from case 4. of Prop. 4 it follows that $\forall \beta \in totdef(\alpha)$ $(totdef(\beta\alpha) \cap \mathcal{F}^k(\emptyset)) \neq \emptyset \wedge \exists \beta \in totdef(\alpha) \mid (totdef(\beta\alpha) \cap (\mathcal{F}^k(\emptyset) \setminus \mathcal{F}^{k-1}(\emptyset))) \neq \emptyset$. This implies that:

- $\forall \beta \in dirdef(\alpha)$ α is defended by $\mathcal{F}^k(\emptyset)$ against β, namely there is an argument γ such that $\gamma\beta\alpha$ or $\gamma\beta$ belongs to $\mathcal{F}^k(\emptyset)$ (in both cases it must also hold $\gamma \in \mathcal{F}^k(\emptyset)$);
- letting $\epsilon = src(\alpha)$, $\forall \beta \in indirdef(\alpha) = dirdef(\epsilon)$ α is defended by $\mathcal{F}^k(\emptyset)$ against β, namely there is an argument γ such that $\gamma\beta\epsilon$ or $\gamma\beta$ belongs to $\mathcal{F}^k(\emptyset)$ (in other words ϵ is defended by $\mathcal{F}^k(\emptyset)$).

In all cases it must also hold $\gamma \in \mathcal{F}^k(\emptyset)$ and for at least one of these elements γ it must hold that either $\gamma\beta\alpha$ or $\gamma\beta\epsilon$ or $\gamma\beta$ belongs to $\mathcal{F}^k(\emptyset) \setminus \mathcal{F}^{k-1}(\emptyset)$.

By the inductive hypothesis, if follows that for any such γ, $argst(\gamma)$ is marked as $\mathbf{in}(i)$ with $i \leq k$ and either $repst(\gamma\beta\alpha)$ or $repst(\gamma\beta\epsilon)$ or $repst(\gamma\beta)$ is marked as $\mathbf{in}(i)$ with $i \leq k$ (again, for at least one of these elements, the mark is exactly $\mathbf{in}(k)$). It follows that $\forall \beta \in totdef(\alpha)$ $repst(\beta\alpha)$ is marked out at an iteration $i \leq k$ and one of these $repst(\beta\alpha)$ is marked out exactly at the iteration k. Hence $repst(\alpha)$ becomes unattacked, and hence is marked **in**, exactly at the iteration $k+1$ and $\alpha \in L_{k+1}$ as desired.

On this basis we obtain one of the main results of the paper.

Theorem 2 *Let* $\mathcal{M}^+ = \langle \mathcal{X}, Q_{\mathcal{M}^+}, q_0, F_{\mathcal{M}^+}, \delta^+ \rangle$ *with* $\alpha \in L(\mathcal{M}^+) \Leftrightarrow \alpha \in \mathcal{X} \cup \mathcal{R}$ *be a* DFA$^+$ *describing the* AFRA*,* $\langle \mathcal{X}, \mathcal{R} \rangle$ *with corresponding* AF $\langle \tilde{\mathcal{X}}, \tilde{\mathcal{R}} \rangle$*. It is possible to construct in polynomial time a* DFA $\mathcal{M}_G = \langle \mathcal{X}, Q_G, q_0, F_G, \delta_G \rangle$ *with* $\alpha \in L(\mathcal{M}_G) \Leftrightarrow \alpha \in GE(\langle \tilde{\mathcal{X}}, \tilde{\mathcal{R}} \rangle)$

Proof. Given Prop. 5, we have only to show that Alg. 1 terminates in polynomial time. We have already commented that the *csplit* operation (l. 4) can be carried out in polynomial time and gives rise to a total number of states $\#Q \leq |Q_{\mathcal{M}^+}| * |\mathcal{X}|$. The **repeat** cycle terminates when $\mathcal{M}_i = \mathcal{M}_{i-1}$, which occurs when no unmarked unattacked states are detected at iteration i. Identifying whether a state q is unattacked requires the following checks (check (ii) only applies to attack states): (i) for any state $p \in state\!-\!out(q)$ is p in F_i? (ii) for any argument $x \in sym - in(q)$ is any defeater of x in F_i?

Check (i) requires at most $\#Q$ constant time operations for each state q, so its complexity in a single iteration of the **repeat** cycle is $O(\#Q^2)$. Check (ii) requires at most $|\mathcal{X}|^2$ constant time operations for each state q, so its complexity in a single iteration of the **repeat** cycle is $O(\#Q * |\mathcal{X}|^2)$.

Given the identification of unattacked states for granted, in a single iteration of the **repeat** cycle:

- at most $\#Q$ mark operation are executed at l. 7;
- at most $\#Q$ checks on membership to $state - in(q) \cap F_i$ are carried out at l. 8 and at most the same number of marking and removal operations are executed at l. 9;
- the **for** cycle at l. 11 is executed at most $|\mathcal{X}|$ times and for each of these iterations at most $\#Q$ marking and removal operations are executed at l. 12.

Noting that the algorithm never adds accepting states, it follows that the number of removals and, hence, the number of iterations of the **repeat** cycle is bounded by $\#Q$. Finally the **for** cycle at l. 15 is executed at most $\#Q$ times.

Summing up, the order of magnitude of the computational complexity of Alg. 1 is determined by checks (i) and (ii) within the **repeat** cycle, which turn out to be respectively $O(\#Q^3) = O(|Q_{\mathcal{M}^+}|^3 * |\mathcal{X}|^3)$ and $O(\#Q * \#Q * |\mathcal{X}|^2) = O(|Q_{\mathcal{M}^+}|^2 * |\mathcal{X}|^4)$. □

7 Conclusions

This paper proposes a methodology and provides some initial results in the largely unexplored field of computing with infinite argumentation frameworks, using as a starting point the possible existence of infinite attacks in the recently introduced AFRA formalism, exemplified by a case of moral dilemma. While other approaches (for instance, Modgil's EAF [13]) may provide a different formalization of this specific example, from a general point of view it is worth noting that the notion of unlimited recursive attacks, as in the AFRA formalism, may encompass infinite attack sequences even with a finite set of arguments. This can be easily seen as a finite alphabet able to describe infinite attack structures.

In fact, the proposal is built on the main idea of drawing correspondences between the specification of argumentation frameworks and well-known notions and results in formal language theory. While there are cases of infinite attacks which can not be represented with formal grammars, deterministic finite automata provide a convenient way to represent infinite attack relations with potential practical use. In particular we show that, with this representation, the problem of computing the grounded extension, which is tractable in the finite case, preserves its tractability in the infinite case. We are already extending this kind of analysis to other "standard" computational problems in abstract argumentation, like checking whether a set is conflict-free, is admissible or is a stable extension. The representation of special reasoning cases, like dilemmas, is an example of motivation for this kind of studies. In a similar spirit, one might consider the representation of dialogues where the repetition of previous moves is allowed: while this is normally forbidden, in order to ensure dialogue termination, the proposed approach might be used to define a sound semantics for some kinds of non-terminating dialogues, which represent the formal counterpart of situations where dialogue participants decide to keep (some of) their positions forever [10, 9].

In the perspective of enlarging its applicability domain, the proposed methodology and techniques could also be applied to other cases of infinite frameworks, either in the context of traditional Dung's AF or in some of its extended versions. In particular, it can be noted that the proposed approach implicitly deals with a family of infinite Dung's AFs since any AFRA with infinite attacks can be translated into a traditional AF with infinite arguments (see Sect. 2). From a more general perspective, one can consider using the DFA representation to specify an infinite set of arguments (so that each accepted word corresponds to an argument) complemented by a compact definition of the attack relation. Just to give an example, one simple option is to state that if both words xw and w are accepted (i.e. both of them represent arguments) then xw attacks w. In this way it is possible, for instance, to represent an infinite chain of attacks with a simple DFA, accepting the words x, xx, xxx, ... A more general option is to specify the attack relations through an expression constructed by a set of operators. A variant of Algorithm 1 could then be devised to compute the grounded extension of this kind of frameworks. A deep investigation of these issues is the subject of ongoing work [3].

References

1. Baroni, P., Cerutti, F., Giacomin, M., Guida, G.: An argumentation-based approach to modeling decision support contexts with what-if capabilities. In: AAAI Fall Symposium. Technical Report SS-09-06. pp. 2–7 (2009)
2. Baroni, P., Cerutti, F., Giacomin, M., Guida, G.: AFRA: argumentation framework with recursive attacks. International Journal of Approximate Reasoning 51(1), 19–37 (2011)
3. Baroni, P., Cerutti, F., Dunne, P.E., Giacomin, M.: Automata for infinite argumentation structures. Technical Report, Department of Computer Science, University of Liverpool, UK (2011)
4. Barringer, H., Gabbay, D., Woods, J.: Temporal dynamics of support and attack networks: From argumentation to zoology. In: Hutter, D., Stephan, W. (eds.) Mechanizing Mathematical Reasoning, Lecture Notes in Computer Science, vol. 2605, pp. 59–98. Springer Berlin / Heidelberg (2005)
5. Bench-Capon, T.J.M.: Persuasion in practical argument using value based argumentation frameworks. Journal of Logic and Computation 13(3), 429–448 (2003)
6. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and N-person games. Artificial Intelligence 77, 321–357 (1995)
7. Dunne, P.E.: Computability Theory – concepts and applications. Ellis–Horwood (1991)
8. Dunne, P.E., Modgil, S., Bench-Capon, T.: Computation in extended argumentation frameworks. In: Proceedings of 19th European Conference on Artificial Inteligence (ECAI 2010). pp. 119–124. Lisbon, P (2010)
9. Gabbay, D., Woods, J.: More on non-cooperation in dialogue logic. Logic Journal of IGPL 9(1), 305–323 (2001)
10. Gabbay, D., Woods, J.: Non-cooperation in dialogue logic. Synthese 127, 161–186 (2001)
11. Gabbay, D.: Semantics for higher level attacks in extended argumentation frames part 1: Overview. Studia Logica 93, 357–381 (2009)
12. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley (1979)
13. Modgil, S.: Reasoning about preferences in argumentation frameworks. Artificial Intelligence 173(9–10), 901–934 (2009)
14. Walton, D., Reed, C., Macagno, F.: Argumentation Schemes. Cambridge University Press, NY (2008)