On the Featured Argumentation Framework

Gianvincenzo Alfano[1,*], Sergio Greco[1], Francesco Parisi[1] and Irina Trubitsyna[1]

[1]*Department of Informatics, Modeling, Electronics and System Engineering (DIMES), University of Calabria, Rende, Italy*

Abstract

We discuss a novel extension of Dung's Argumentation Framework (AF) called *Featured AF* (FAF), where each argument has associated a set of features expressed by means of unary and binary facts [1]. Queries in FAF can be expressed by means of a conjunctive relational calculus formula, evaluated over the FAF extensions. We also discuss the *Extended FAF (EFAF)*, a FAF extension where a first-order logic formula is used for reasoning over 'feasible' subframeworks that satisfy the formula and *minimally* differ from the original framework.

Keywords

Formal Argumentation, Abstract Argumentation, Knowledge Bases

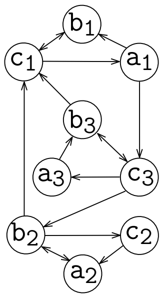
Introduction

Various proposals have been made to extend the Dung's framework with the aim of better modeling the knowledge to be represented. The extensions include Bipolar AF [2, 3], AF with recursive attacks and supports [4, 5, 6], Dialectical framework [7], AF with preferences [8, 9, 10] and constraints [11, 12, 13, 14], as well extensions for representing uncertain information. For the representation of uncertain information, two main extensions have been proposed: *incomplete AF* (iAF), where arguments and attacks may be uncertain [15, 16], and *probabilistic AF* (PrAF), where arguments and attacks are associated with a probability [17, 18, 19, 20, 21].

Example 1. Consider a scenario in which three air pollution experts, Alice (a), Bob (b), and Carl (c), are asked to propose short-, medium-, and long-term solutions to the problem. The experts have different general opinions: Alice supports industrial pollution regulation policies, Bob supports traffic limitation policies, and Carl promotes changes in human habits. Each expert expresses opinions for the short, medium, and long term, represented by the arguments x_1, x_2, x_3, respectively, where x denotes the first letter of the experts' name. These 9 arguments are summarized in Figure 1, while the following attacks can be derived from considerations made by the three experts.
Short-term: (a_1, b_1), (b_1, c_1), (c_1, a_1), (c_1, b_1); Medium-term: (a_2, b_2), (b_2, c_2), (c_2, a_2), (b_2, a_2);
Long-term: (a_3, b_3), (b_3, c_3), (c_3, a_3), (c_3, b_3); Cross-timeframes: (a_1, c_3), (c_3, b_2), (b_2, c_1), (b_3, c_1).
The derived AF Λ (Figure 1) has two stable/preferred extensions: E_1=$\{a_1, b_2, a_3\}$ and $E_2 = \{c_1, c_2, c_3\}$.

The previous example shows how possible air pollution policies can be defined using AF under stable (and preferred) semantics. However, by modeling the problem by means of AF, important information that can be very useful is neglected. In our example, useful information about the text associated with the argument, the author (and her sex and age), and the type of policy (short-, medium-, long-term) is disregarded. Generally, an argument is associated with a sentence with a proper meaning. This means that, in adopting the abstraction of AF, a lot of relevant information, such as the argument's text, the author's id and gender, the date it was introduced, the topic (e.g. what the argument is about), the polarity (e.g. whether the sentence conveys a positive, negative or neutral sentiment), the polarity rating, the sentence style (e.g. veracity, sarcasm, irony), and many others, are lost.

AI[3] 2025: 9th Workshop on Advances in Argumentation in Artificial Intelligence, September 13, 2025, Rende, Italy
*Corresponding author.
g.alfano@dimes.unical.it (G. Alfano); greco@dimes.unical.it (S. Greco); fparisi@dimes.unical.it (F. Parisi); i.trubitsyna@dimes.unical.it (I. Trubitsyna)
0000-0002-7280-4759 (G. Alfano); 0000-0003-2966-3484 (S. Greco); 0000-0001-9977-1355 (F. Parisi); 0000-0002-9031-0672 (I. Trubitsyna)



a_1: Impose immediate fines and sanctions on industries exceeding pollution limits;
a_2: Enforce stricter industrial regulations with compliance mechanisms;
a_3: Mandate the adoption of green technologies to achieve sustainable industrial practices;
b_1: Implement congestion charges and restrict access for high-polluting vehicles;
b_2: Expand public transportation to reduce private vehicle use;
b_3: Redesign cities to prioritize pedestrians, cyclists, and electric vehicles;
c_1: Launch awareness campaigns to encourage reduced car usage and energy conservation;
c_2: Integrate environmental education into schools and workplaces;
c_3: Promote societal norms through incentives for eco-friendly behavior and penalties for unsustainable actions.

Figure 1: AF Λ of Example 1.

In [1], the underlying belief is that these features should be made explicit and appropriately used. For these reasons, an extension of AF called *Featured Argumentation Framework (FAF)* has been proposed, where arguments might have associated features expressed by means of (first-order unary and binary) facts. Thus, the knowledge is represented through an underlying AF and a set of facts representing the features associated with arguments and with the AF's topology.

Example 2 (cont'd)**.** Each argument could have associated features such as *author* (e.g. *author* (a_1, *Alice*)), *text* (e.g. *text* (c_2, *"integrate environmental education into schools and workplaces"*)), supported *policy*, (e.g. *policy* (b_1, *traffic_limitation*)), *term* time (e.g. *term* (c_3, *long*)). □

In such a context, users pose queries consisting of conjunctive relational calculus formulae such as checking whether there exists an extension containing only arguments concerning policies proposed by the same expert (E_2 is an extension of such kind in Example 1). Herein, a query is evaluated w.r.t. the extensions of the underlying AF, each consisting of a subset of arguments with the associated features.

Featured AF

We now discuss the Featured Argumentation Framework (FAF), an extension of Dung's AF that consists of adding features associated with the arguments [1]. We assume the reader is familiar with basic notions underlying AF.

Syntax. We assume to have distinct countable sets of arguments $\mathbb{A}$, constants $\mathbb{D}$, variables $\mathbb{V}$, and (unary and binary) *base* predicates $\mathbb{P}_b$. Constants can be either natural numbers or alphanumeric strings starting with a lower-case letter, whereas variables are denoted by alphanumeric strings starting with an upper-case letter. We also assume to have the *predefined (or built-in)* unary predicate arg. $\mathbb{P}$ denotes the set of all predicates (base and built-in). Arguments, constants and variables are terms. An atom is of the form $p(a, b)$ (resp. $p(a)$), where p is a binary (resp. unary) predicate symbol and a, b are terms. Unary atoms take values from $\mathbb{A}$, (base) binary atoms take values from $\mathbb{A} \times \mathbb{D}$. A variable-free base atom is called a *fact*. Given a set A of arguments, Arg_A is used to denote the set of built-in facts $\{arg(a) \mid a \in A\}$.

Definition 1 (FAF syntax [1])**.** *A Featured AF (FAF) is a triple* $\Omega = \langle A, R, F\rangle$ *where* $\langle A, R\rangle$ *is an AF and* F *is a finite set of base facts.*

For an FAF $\langle A, R, F\rangle$, the set of base and built-in facts will be denoted by $F^A = F \cup Arg_A$.

Semantics. Given a set of (base and built-in) facts $\mathcal{F}$ and a set of arguments S, we use $\mathcal{F}_{\downarrow S} = \{p(a) \mid p(a) \in \mathcal{F} \wedge a \in S\} \cup \{p(a, b) \mid p(a, b) \in \mathcal{F} \wedge a \in S\}$ to denote the projection of $\mathcal{F}$ over S. That is, $\mathcal{F}_{\downarrow S}$ contains only the facts in $\mathcal{F}$ referring to the arguments in S. For any AF $\langle A, R\rangle$ and AF semantics σ (e.g. grounded, complete, stable, and preferred), $\sigma(\langle A, R\rangle)$ denotes the set of σ-extensions for $\langle A, R\rangle$.

Definition 2 (FAF Semantics [1])**.** *Given an FAF* $\Omega = \langle A, R, F\rangle$ *and a semantics* σ*, a* σ*-extension for* Ω *is any set* $E \in \sigma(\langle A, R\rangle)$*.*

The set of σ-extensions for Ω is denoted by $\sigma(\Omega)$. Thus, $\sigma(\langle A, R, F\rangle) = \sigma(\langle A, R\rangle)$.

A query is a conjunctive relational calculus query defined over the database F^A, that is, it only uses the predicate symbols contained in F^A. A query is said to be boolean if all variables are (existentially) quantified. For instance, the boolean query $arg(a)$ checks the acceptance of argument a.

Definition 3 (Query acceptance [1]). *For any FAF $\Omega = \langle A, R, F\rangle$, semantics σ, and boolean query q, we say that q is (i) credulously accepted if there exists a set $E \in \sigma(\Omega)$ such that $F^A_{\downarrow E} \models q$ (where $F^A_{\downarrow E}$ denotes the set $(F^A)_{\downarrow_E}$) and (ii) skeptically accepted if for every set $E \in \sigma(\Omega)$, $F^A_{\downarrow E} \models q$ holds.*

Example 3 (cont'd). Consider the FAF $\Omega = \langle A, R, F\rangle$ obtained from the AF $\langle A, R\rangle$ of Example 1, where F contains the following base facts $\{author(a_i, \mathit{Alice}), author(b_i, \mathit{Bob}), author(c_i, \mathit{Carl}) \mid i \in [1, 3]\}$ $\cup$ $\{term(x_1, \mathit{short}), term(x_2, \mathit{medium}), term(x_3, \mathit{long}) \mid x \in \{a, b, c\}\}$. Assume we are interested to know whether there are stable extensions containing at least one short-term argument from Alice, thus we interested in the answer to the query $q = \exists X.author(X, \mathit{Alice}) \land\ term(X, \mathit{short})$. The answer to q is "yes" as there exists a stable-extension $E_1 = \{a_1, b_2, a_3\}$ for Ω and $F^A_{\downarrow E}$ contains the atoms $author(a_1, \mathit{Alice})$ and $term(a_1, \mathit{short})$. Although q is credulously accepted under stable semantics, it is not skeptically accepted, as there exists a stable-extension $E_2 = \{c_1, c_2, c_3\}$ for Ω s.t. $F^A_{\downarrow E'} \not\models q$.

Extended FAF

Consider the running example, and assume now the case where there are two politicians, the former is interested only in short-term policies, the latter is interested only in policies regarding pollution regulations and traffic limitations. For reasoning about the first context (concerning short-term policies), we should focus on the subframework containing only the arguments a_1, b_1 and c_1, whereas for the second context we should consider the subframework containing only the arguments whose only policies are *pollution_regulation* and *traffic_limitation*.

To this end, an extension of FAF called *Extended Featured Argumentation Framework (EFAF)* has been proposed [1]. EFAF extends FAF with a FOL formula (also called constraint) to be satisfied. The FOL formula is used to extract FAF *subframeworks* (subframeworks for short), that are subsets of the original FAF (i.e. the original EFAF without the constraint), that satisfy the constraints and differ from the original FAF by a minimal set or a minimum number of arguments and attacks.

Example 4 (cont'd). The subframework containing only short-term policies can be defined by means of the FOL formula $\forall X.term(X, \mathit{short})$, whereas the subframework considering only arguments denoting policies regarding both pollution regulation and traffic limitation can be defined by means of the FOL formula $\forall X.policy(X, \mathit{pollution\_regulation}) \lor policy(X,\ \mathit{traffic\_limitation})$. On the other side, the formula $\forall X.policy(X, \mathit{pollution\_regulation}) \lor \forall X.policy(X,\ \mathit{traffic\_limitation})$ allows determining two alternative subframeworks.

In a multi-agent context, the FAF encodes shared knowledge, while each constraint reflects an individual agent's perspective. The FOL formula can be viewed as an integrity constraint that guarantees that available arguments and features correctly model the outside world.

Syntax. We first recall the concept of inclusion for FAF. Given two FAFs Ω and Ω', we write $\Omega' \sqsubseteq \Omega$ (resp. $\Omega' \sqsubset \Omega$) if Ω' can be obtained from Ω through the deletion of a possibly empty (resp. non-empty) set of arguments (as well as the related base facts) and attacks. That is, $\Omega' \sqsubseteq \Omega$ iff $A' \subseteq A$, $R' \subseteq R$, and $F' = F_{\downarrow_{A'}}$; moreover, $\Omega' \sqsubset \Omega$ iff $\Omega' \sqsubseteq \Omega$ and Ω' is not equal to Ω.

Given an FAF $\Omega = \langle A, R, F\rangle$, we assume to also have the built-in binary predicate att, denoting attacks among arguments, and the *derived* binary predicates $path$, $even\text{-}path$ and $odd\text{-}path$ denoting paths over the argumentation graph $\langle A, R\rangle$; all these predicates take values from $A \times A$. Hereinafter, we use Att_R to denote the set $\{att(a, b) \mid (a, b) \in R\}$, and $\mathbb{P}$ to denote the set of all predicates (base, built-in, and derived ones). We use λ to denote the first-order language using base, built-in and derived predicates, and terms (i.e. arguments, constants and variables).

Definition 4 (EFAF Syntax [1]). *An extended FAF (EFAF) is a tuple $\Delta = \langle A, R, F, C\rangle$ where $\langle A, R, F\rangle$ is an FAF and C is a first-order formula defined over λ.*

$F^{AR} = F \cup Arg_A \cup Att_R$ is used to denote the set containing base feature facts and built-in facts describing the topology of the underlying graph.

Semantics. We start by defining the satisfaction of the formula C with respect to the underlying FAF $\langle A, R, F \rangle$. A base or built-in fact $p(a, b)$ (resp. $p(a)$) is satisfied w.r.t. a EFAF $\Delta = \langle A, R, F, C \rangle$ if it occurs in F^{AR}. Moreover, a derived fact $path(a, b)$ (resp. $even\text{-}path(a, b)$, $odd\text{-}path(a, b)$) is satisfied w.r.t. Δ if there exists a path (resp. even simple path, odd simple path) from a to b in the underlying AF $\langle A, R \rangle$. The role of the logical formula C in Δ is to represent a constraint (or a set of constraints) to be satisfied by F^{AR}. Intuitively, if the logical formula C is not satisfied, then the underlying FAF should be revised by computing subframeworks, that is, by (minimally) modifying the topology of the AF through the deletion of arguments and attacks, so that the formula is satisfied. Clearly, when deleting arguments, related features and attacks are deleted as well. In the following, we assume that the formula C is satisfiable, that is that there exists an FAF $\langle A, R, F \rangle$ such that $F^{AR} \models C$. The semantics of EFAF is based on the notion of subframework, which is formally recalled next.

Definition 5 (Subframework [1])**.** *Given an EFAF $\Delta = \langle A, R, F, C \rangle$, a set-subframework (resp. cardinality-subframework) for Δ is an FAF $\langle A', R', F' \rangle \sqsubseteq \langle A, R, F \rangle$ such that, (i) $F^{AR}_{\downarrow A'} \models C$, and (ii) there is no FAF $\langle A'', R'', F'' \rangle \sqsubseteq \langle A, R, F \rangle$ such that $F^{AR}_{\downarrow A''} \models C$ and $\langle A', R', F' \rangle \sqsubset \langle A'', R'', F'' \rangle$ (resp. $|A'| + |R'| < |A''| + |R''|$).*

Thus, a set-subframework (s-subframework for short) is obtained by deleting a minimal set of arguments and attacks, whereas a cardinality subframework (c-subframework) is obtained by deleting a minimum number of arguments and attacks. Observe that, in computing subframeworks, feature facts are implicitly deleted through the deletion of the related arguments. This is in line with repairing strategies exploited in databases, where the inconsistency of an attribute leads to the deletion of an entire tuple [22].

Example 5. Consider the EFAF $\Delta = \langle A, R, F, C \rangle$ obtained from the FAF $\langle A, R, F \rangle$ of Example 3, where the FOL formula $C = \forall X.term(X, \mathit{short})$, captures the subframeworks containing only short-term policies. The unique subframework (under set or cardinality semantics) is obtained by deleting every argument x such that $term(x, \mathit{short})$ is false, and its unique stable extension is $\{c_1\}$.

Assume we want to fix exactly one term policy (either short, medium, or long), independently of the actual term's value. The FOL formula $C = \forall X, Y, Z, Z'\ term(X, Z) \wedge term(Y, Z') \Rightarrow Z = Z'$ allows to identify three s-subframeworks obtained by keeping only the arguments sharing the same term's value.

Problems and Complexity. Observe that a given EFAF may admit zero, one, or multiple subframeworks. Subframework existence, verification, and acceptance problems can be defined analogously to those defined for iAF [15, 16]. In fact, roughly speaking, the main differences with respect to iAF is that subframeworks correspond to completions which must satisfy a minimality criteria (either set or cardinality subframeworks semantics). Regarding complexity, it has been shown in [1] that the complexity of acceptance problems in FAF and AF coincides. Differently, the complexity of acceptance problems in EFAF is strictly harder than the AF counterpart.

Relationships with incomplete AF. Notice that EFAF without base predicates is similar to correlated/-constrained incomplete AF [23, 24], though the formulae defined in those works are propositional and allow only built-in predicates. However, the aim is different, as in EFAF the interest is in dealing with subframeworks (which minimally differ from the original framework), whereas in the above-mentioned works all *completions* are first-class citizens. In [1], it has been shown that (correlated/constrained) iAF is a special class of EFAF as every (correlated/constrained) iAF can be rewritten into an extensions-equivalent EFAF, modulo meta-elements (i.e., arguments/attacks added in the rewriting).

Acknowledgments. We acknowledge financial support from PNRR MUR projects FAIR (PE0000013) and SERICS (PE00000014), project Tech4You (ECS0000009), and MUR project PRIN 2022 EPICA (H53D23003660006).

Declaration on Generative AI. The author(s) have not employed any Generative AI tools.

References

[1] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, Featured argumentation framework: Semantics and complexity, in: Proc. of IJCAI, 2025, p. (to appear).

[2] F. Nouioua, V. Risch, Argumentation frameworks with necessities, in: Proc. of SUM Conf., 2011.

[3] S. Villata, G. Boella, D. M. Gabbay, L. W. N. van der Torre, Modelling defeasible and prioritized support in bipolar argumentation, Ann. Math. Artif. Intell. 66 (2012).

[4] A. Cohen, S. Gottifredi, A. J. Garcia, G. R. Simari, An approach to abstract argumentation with recursive attack and support, J. Appl. Log. 13 (2015) 509–533.

[5] C. Cayrol, J. Fandinno, L. F. del Cerro, M. Lagasquie-Schiex, Structure-based semantics of argumentation frameworks with higher-order attacks and supports, in: Proc. of COMMA Conf., 2018, pp. 29–36.

[6] G. Alfano, A. Cohen, S. Gottifredi, S. Greco, F. Parisi, G. R. Simari, Credulous acceptance in high-order argumentation frameworks with necessities: An incremental approach, Artif. Intell. 333 (2024) 104159.

[7] G. Brewka, H. Strass, S. Ellmauthaler, J. P. Wallner, S. Woltran, Abstract dialectical frameworks revisited, in: Proc. of IJCAI Conf., 2013, pp. 803–809.

[8] L. Amgoud, C. Cayrol, On the acceptability of arguments in preference-based argumentation, in: Proc. of UAI Conf., 1998, pp. 1–7.

[9] S. Modgil, H. Prakken, A general account of argumentation with preferences, Artif. Intell. 195 (2013) 361–397.

[10] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, Abstract argumentation framework with conditional preferences, in: Proc. of AAAI Conf., 2023, pp. 6218–6227.

[11] S. Coste-Marquis, C. Devred, P. Marquis, Constrained argumentation frameworks, in: Proc. of KR Conf., 2006, pp. 112–122.

[12] O. Arieli, Conflict-free and conflict-tolerant semantics for constrained argumentation frameworks, J. Appl. Log. 13 (2015) 582–604.

[13] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, Preferences and constraints in abstract argumentation, in: Proc. of International Joint Conference on Artificial Intelligence, 2023, pp. 3095–3103.

[14] G. Alfano, S. Greco, D. Mandaglio, F. Parisi, I. Trubitsyna, Abstract argumentation frameworks with strong and weak constraints, Artif. Intell. 336 (2024) 104205.

[15] D. Baumeister, D. Neugebauer, J. Rothe, H. Schadrack, Verification in incomplete argumentation frameworks, Artif. Intell. 264 (2018) 1–26.

[16] D. Baumeister, M. Järvisalo, D. Neugebauer, A. Niskanen, J. Rothe, Acceptance in incomplete argumentation frameworks, Artif. Intell. (2021) 103470.

[17] P. M. Dung, P. M. Thang, Towards (probabilistic) argumentation for jury-based dispute resolution, in: Proc. of COMMA Conf., 2010, pp. 171–182.

[18] H. Li, N. Oren, T. J. Norman, Probabilistic argumentation frameworks, in: Proc. of TAFA Workshop, 2011, pp. 1–16.

[19] A. Hunter, Some foundations for probabilistic abstract argumentation, in: Proc. of COMMA Conf., 2012, pp. 117–128.

[20] G. Alfano, M. Calautti, S. Greco, F. Parisi, I. Trubitsyna, Explainable acceptance in probabilistic and incomplete abstract argumentation frameworks, Artif. Intell. 323 (2023) 103967.

[21] B. Fazzinga, S. Flesca, F. Furfaro, Abstract argumentation frameworks with marginal probabilities, in: Proc. of IJCAI, 2022, pp. 2613–2619.

[22] M. Arenas, L. E. Bertossi, J. Chomicki, Consistent query answers in inconsistent databases, in: Proc. of PODS Conf., 1999, pp. 68–79.

[23] B. Fazzinga, S. Flesca, F. Furfaro, Reasoning over argument-incomplete aafs in the presence of correlations, in: Proc. of IJCAI Conf., 2021, pp. 189–195.

[24] J. Mailly, Possible controllability of control argumentation frameworks, in: Proc. of COMMA Conf., 2020, pp. 283–294.