Most-Probable: A New Argumentation Semantics through Optimization*

Davide Di Pierro[1,*,†], Stefano Ferilli[1,*,†]

[1]Università degli Studi di Bari Aldo Moro

Abstract

Reasoning in Abstract Argumentation is the activity of understanding subsets of arguments that resolve conflicts or can be used as a justification for a thesis. Several semantics have been proposed so far to identify subsets of arguments showing properties based on self-consistency or defence against attackers. Real-world scenarios and peculiarities of many problems gave rise to the introduction of more complex frameworks that also specify degrees of certainty of arguments and attacks. However, semantics barely take into account the fuzziness of these "beliefs" properly. In this work, we propose a strategy to compute arguments' beliefs based on the initial degrees of certainty of arguments and propose a new semantics, *most-probable*, that, starting from a target argument we aim to justify, captures the set of admissible arguments containing the target argument which maximizes the minimum belief of the whole set. The introduction of this new semantics can facilitate the understanding of the consequences of supporting a claim, neglecting those that may result in low belief.

Keywords

Argumentation, Logic Programming, Answer Set Programming, Optimization

1. Introduction

The field of argumentation can be traced back to what the philosopher Pollock in [1] defined as "defeasible reasoning", which is a branch of nonmonotonic reasoning where conclusions and justifications can be retracted once new information is available. Information is split into arguments, which are conflicting facts available as knowledge. Currently, the theory of argumentation is still also based on Dung's formalization [2, 3]. The *attack* relationship is the only one available in the original version and no numerical information was provided. Generally speaking, the attacker of the attacker is seen as a supporter. This formalization is called *argumentation framework* and is naturally represented as a graph in which *arguments* are represented with nodes and the *attacks* as directed edges from the attacker to the attacked argument. There is the possibility of introducing cycles in the attacks (which is also common in everyday life), or even arguments attacking themselves, even though this scenario is not usually considered. Given the generality of the framework, this formalization is still the most common in the field.

Despite the simplicity of the framework, there is great potential and much demand from the world, especially in the field of law where it goes without saying that for each case there are conflicting facts that need to be sorted out in order to find the most plausible set of claims. An overview of the interconnection between the two fields is provided by Prakken et al. [4].

The representation of conflicting arguments is not an end in itself. Mining arguments represents the possibility of studying whether claims are somehow supported. To this extent, claims are proofs themselves, used as justification for other claims. In principle, there cannot be a general approach to satisfy the needs of one to shed light on the facts, but it is strictly dependent on *how* we consider a claim to be supported. For this reason, different semantics are available, some more specific than others, that can capture arguments with more specific properties in the abstract framework.

8th Workshop on Advances in Argumentation in Artificial Intelligence, November 27, 2024, Bolzano, Italy

*Corresponding author.
†These authors contributed equally.
davide.dipierro@uniba.it (D. Di Pierro); stefano.ferilli@uniba.it (S. Ferilli)
0000-0002-8081-3292 (D. Di Pierro); 0000-0003-1118-0601 (S. Ferilli)

In order to better describe real-world scenarios, a certain degree of uncertainty *should* be introduced in the matter of discourse. Even though Dung's formalization is still the most used, extensions including probabilities, beliefs or fuzziness are available. In the realm of model representation, many techniques are borrowed from the Knowledge Representation and Reasoning field. To this extent, logic programming frameworks provide comprehensive representation tools and strategies to store, manage and compute argumentation semantics. They are particularly suited for this scope given their interpretable nature, symbolic representation and textual (rather than numeric) information. These frameworks have already been studied to support uncertainty with plenty of solutions available for dealing with probabilities, fuzziness or degree of beliefs. In some languages, nonmonotonic reasoning is not inherently available but techniques are possible to add or retract information. Prolog [5], one of the most famous logic programming languages, can represent arguments and attacks with propositional logic (known as *facts*), and reason with them with first-order logic, with the introduction of *rules*. A more recent framework, Answer Set Programming [6] (ASP), can represent arguments and attacks in the same way but embeds in rules more complex operators that facilitate or empower traditional predicate logic rules.

In this work, we propose a strategy to deal with a degree of beliefs and a new semantics focused on a *target* argument. Specifically, we suppose a given knowledge of arguments is available with previous beliefs (regardless of the others). After arguments are put into correlation with the attacks, the likelihood of arguments *should* vary, in correlation with the beliefs of the attackers. After updating beliefs, the *most-probable* extension will be computed. It can be considered as a variant of a very well-known one (shortly to be described) with the introduction of an optimisation filter to rule out arguments that reduce the overall likelihood of the set of *accepted* facts. The overall solution is provided with full first-order logic frameworks with (mainly) Prolog for the representation of arguments and attacks, and ASP for the *most-probable* extension given its optimization functions. We see a natural use in the field of law when trying to identify which are all the facts consistent with a claim (e.g. "A is guilty") with the highest possible belief.

This manuscript is structured as follows: Section 2 introduces theory on argumentation and semantics, Section 3 specifies the proposed strategy and semantics, Section 4 provides technical details about implementation of the proposed setting, Section 5 lists related works in the field, and finally Section 6 makes evidence of consequences and concludes the manuscript.

2. Background

Although this is not intended to be a comprehensive review of the literature in the argumentation field, we provide basic definitions and concepts for this work. Definitions mainly have roots from the first Dung's formalization and the proposed basic semantics.

Definition 1 An *Argumentation Framework (AF)* is a pair $\langle Args, Att \rangle$ where $Args$ is a (finite) set of arguments and $Att \subseteq Args \times Args$ the attack relationship. If $\langle A, B \rangle \in Att$ we say that argument A *attacks* argument B (or that A is an attacker of B). For each $A \in Args$, we denote as $A^- = \{ B \in Args : \langle B, A \rangle \in Att \}$ the set of all *attackers* of A, and as $A^+ = \{ B \in Args : \langle A, B \rangle \in Att \}$ the set of all *attacked* by A. In this scope, we neglect self-attacking arguments, so $A \in Args \Longrightarrow A \notin A^- \ (A^+)$. The concept of attack can also be extended to a set of arguments. Given $S \subseteq Args$, $A \in Args$ *attacks* S iff $exists\, B \in S$ such that A attacks B. Conversely, a set $S \subseteq Args$ attacks $A \in Args$ iff $\exists B \in S$ such that B attacks A. Figure 1 shows an example of an argumentation graph.

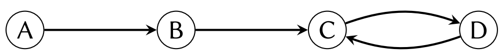

Figure 1: An example of an Argumentation Framework with arguments A, B, C and D.

Following Li et al. [7], we provide the following definition.

Definition 2 Let an $AF\ G = \langle Args, Att \rangle$, a *Probabilistic Argumentation Framework (PAF)* $\langle Args, Att, PArgs, PAtt \rangle$ is a 4-tuple in which $Args$ and Att are defined as above, $PArgs : Args \rightarrow]0, 1]$ and $PAtt : Att \rightarrow]0, 1]$ functions respectively indicating the likelihood of arguments and attacks.

It is often the case that the terms "probability" and "belief" are confused. They both represent a degree of certainty but have different meanings and implications, as indicated by Fagin et al. [8]. Figure 2 shows an argumentation graph with probabilities attached to arguments and attacks.

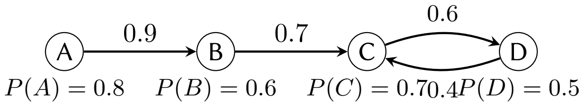

Figure 2: An example of a Probabilistic Argumentation Framework.

Slightly modifying Definition 1, we can generalise the probabilistic framework with a generic *weighted* framework. From Bistarelli et al. [9] we provide the following definition.

Definition 3 Let an $AF\ G\langle Args, Att \rangle$, a *Weighted Argumentation Framework* $\langle Args, Att, WAtt \rangle$ is a triple in which $Args$ and Att are defined as above, and $WAtt : Att \rightarrow \mathbb{R}_{>}$ a function indicating the weight of attacks.

Differently from probabilities, the meaning of weights does not reflect how probable is the attack, but how *much* the arguments attack. Figure 3 shows an argumentation graph with weights attached to attacks.

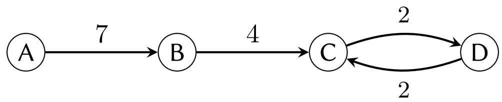

Figure 3: An example of a Weighted Argumentation Framework.

The Argumentation Framework and its variants can be enriched with the *support* relationship, the opposite of the attack. In some contexts, it is advantageous to add support relationships, but these considerations are beyond the scope of this work.

Given an AF, we are interested in collecting sets of arguments that can be generally considered *acceptable*. The condition of being acceptable depends on the *semantics*.

Definition 4 Let an $AF\ G = \langle Args, Att \rangle$ be an AF, an *extension-based semantics* S associates AF with a subset of 2^{Args}, denoted by $\varepsilon_S(AF)$. Every subset of 2^{Args} can be considered a distinct semantics (even $\emptyset$), but only some of them are convenient for acceptability. The basic condition for a set of arguments to be generally acceptable is to show the *conflict-free* condition;

Definition 6 Let an $AF\ G = \langle Args, Att \rangle$ be an AF, a set of arguments $S \subseteq Args$ is *conflict-free* iff $\nexists \langle A, B \rangle \in S$ such that $\langle A, B \rangle \in Att$. The set of all conflict-free extensions is indicated as cf(G).

Referring to the example in Figure 1, in Figure 4 two ways of grouping arguments into conflict-free sets are shown.

Figure 4: Two instances of the same argumentation graph. In the left picture, A is put together with C, and hence B and D can also be collected; in the right picture, A is put together with D, and then B and C must be separated.

This semantics guarantees to obtain groups of arguments at least *consistent* with each other. Many semantics have a hierarchical structure, from the least restrictive to the most. The conflict-free is the minimal result for acceptability. More interesting semantics start from the conflict-free one. Note that $\emptyset$ is always conflict-free, and, with the assumption of non-self-attacking arguments, singleton sets are conflict-free as well, but this result is not practically interesting.

For distinguishing some semantics it is necessary to define the concept of *defense*.

Definition 5 Let an AF $G = \langle Args, Att \rangle$, $S \subseteq Args$, and $\alpha \in Args$, S *defends* α iff $\forall \gamma \in Args \, \langle \gamma, \alpha \rangle \in Att \Longrightarrow \exists B \in S$ such that $\langle B, \gamma \rangle \in Att$.

Definition 7 Let an AF $G = \langle Args, Att \rangle$ be an AF, a set of arguments $S \subseteq Args$ is *admissible* iff S is conflict-free and $\forall \gamma \in Args \, \gamma$ attacks $S \Longrightarrow S$ attacks γ. In other words, S defends every $A \in S$. When this happens, we say that S is able to defend itself. The set of all admissible extensions is indicated as adm(G). In Figure 4 three pairs of arguments are conflict-free: $\{A, C\}$, $\{B, D\}$, $\{A, D\}$, but $\{B, D\}$ is not admissible since A attacks B but neither B nor D attacks A; hence, it cannot defend itself. Figure 5 shows the two possible pairs of arguments, white nodes cannot be part of any admissible set.

Figure 5: Two instances of the same argumentation graph. In the left picture, A and C are admissible, but not B and D, even alone; in the right picture, A and D are admissible, and then B and C cannot be part of any admissible set.

In general, we are not only interested in admissibility but also in including as many arguments as possible in order to make the discovery process more interesting.

Definition 8 Let an AF $G = \langle Args, Att \rangle$ be an AF, a set of arguments $S \subseteq Args$ is a *complete* extension iff S is admissible and $\forall \alpha \in Args \; S$ defends $\alpha \Longrightarrow \alpha \in S$. The set of all complete extensions is indicated as comp(G).
The completeness of semantics acts as a means of spreading the defence relationship. Starting from a singleton, in order for the set to be complete, it must include all the arguments it defends, and then progressively add all the others defended by the ones already included in the set.
Having collected all complete extensions, we can collect all the arguments available in every complete extension by intersection. Here, we define *grounded* semantics.

Definition 9 Let an AF $G = \langle Args, Att \rangle$ be an AF, a set of arguments $S \subseteq Args$ is a *grounded* extension iff S is complete and $\nexists T \in$ comp(G) such that $t \subset S$. The grounded extension is the minimal set of arguments belonging to the complete extensions. It is unique and always exists.
In general scenarios, it may be preferable to select extensions that guarantee as many arguments as possible. To this extent, we can define *preferred* extensions.

Definition 10 Let an AF $G = \langle Args, Att \rangle$ be an AF, a set of arguments $S \subseteq Args$ is a *preferred* extension iff S is admissible and $\nexists T \in$ adm(G) such that $S \subset T$. The set of all preferred extensions is indicated as pref(G).

Finally, when computing extensions, it happens that not all the arguments outside preferred extensions are attacked. These are the so-called "undecided" arguments. *Stable* extensions split the set of arguments between arguments in the extension and attacked arguments.

Definition 11 Let an AF $G = \langle Args, Att \rangle$ be an AF, a set of arguments $S \subseteq Args$ is a *stable* extension iff S is conflict-free and $\forall \alpha \in Args \; \alpha \notin S \Longrightarrow S$ attacks α. The set of all stable extensions is indicated as stb(G).

3. Probabilistic Update and Most-Probable Extension

Given the variety of argumentation models and semantics, we define here our settings for the work. As previously said, research on argumentation semantics under uncertainty is still an underrepresented line of research. We start from a small variety of the PAF defined above. Specifically, we suppose probabilities only on arguments. Probabilities come under the assumption of *non-additivity*, meaning that it is not impossible that $\Delta \not\models \alpha$ and $\Delta \not\models \neg\alpha$. Attacks always trigger with the same "intensity". Probabilities of arguments may come from several external sources. In general, statistical approaches come into play when providing an initial estimation of a phenomenon. In this setting, probabilities of arguments do not take into account correlation with other arguments. Albeit this assumption may not hold in real scenarios, it facilitates an initial approximation of probabilities, which would be extremely demanding otherwise since statistical information of events barely can take into account all surrounding conditions.

Formally, we can define our version of PAF, namely *Simplified PAF (SPAF).*

Definition 12 Let an $AF\ G = \langle Args,\ Att\rangle$, a *Simplified Probabilistic Argumentation Framework (SPAF)* $\langle Args,\ Att,\ PArgs\rangle$ is a triple in which $Args$ and Att are defined as above, $PArgs : Args \rightarrow]0,\ 1]$, a function indicating the likelihood of arguments.

Figure 6 shows an example of SPAF.

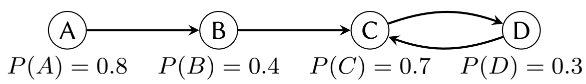

Figure 6: An example of a Simplified PAF.

Note that probabilities of arguments cannot be compared with Kolmogorov Axioms [10] because we do not know in advance whether two arguments are independent. Even C and D in the example above cannot be considered mutually exclusive given possible flaws in the statistical process. The process of attack creation needs an expert in the field who, with the help of prior knowledge, can design a complete SPAF. Conversely, retrieving probabilities of attacks may present some unsound scientific processes or require too accurate domain knowledge.

Probabilities of arguments are "blind" with respect to other arguments and attacks. For this reason, we need to update the probabilities of arguments, based on attacks and the initial probability of arguments. As a general rule, we expect the updated probability inversely proportional to the probability of the attacker and the number of attackers.
The update should consider using a hyperparameter $\alpha \in]0,\ 1]$ to establish the pace at which values change. The hyperparameter is an indicator of how much attacks should impact the acceptability of attacked arguments.

The proposed formula is

$$P'(A) = P(A) \cdot \prod_{\substack{\gamma \in Args \\ \langle \gamma, A\rangle \in Att}} 1 - \alpha \cdot P(\gamma)$$

where $P'(A)$ represents the probability of A after being attacked by all its attackers. The formula presents nice properties but has also some limitations or some assumptions. Nice properties are:

- multiple attacks influence exponentially the updated belief, for this reason, a low α may bring less severe changes.
- the product $1 - \alpha \cdot P(\gamma)$ is always between 0 and 1; hence, the updated value keeps its original interpretation.
- the initial belief of an argument is an upper bound for the update since multiplications of factors between 0 and 1 only reduce the number.

- many weak (low P(γ)) attackers are less influential than few strong (high P(γ)) attackers.

Some limitations are:

- if an attacker has probability 1, only with $\alpha = 1$ the attack has the maximum effect (the updated probability of the attacked is 0).
- the function is nonlinear, meaning that small changes have large effects. This is mitigated by the fact that we are more interested in relative probabilities among arguments rather than absolute numbers.
- it is assumed attacks are independent, which is in general false but neglected in many contexts given the complexity of the dependencies.

We report in Figure 7 two argumentation graphs, the first one after updating probabilities with $\alpha = 0.5$ and the second with $\alpha = 0.9$.

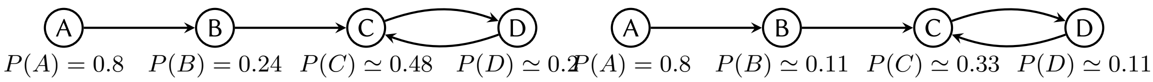

Figure 7: Two instances of probabilistic update in argumentation graph. In the left picture, $\alpha = 0.5$; in the right one, $\alpha = 0.9$.

Note that the whole discussion with probabilities does not change if we deal with generic "degrees of belief". We chose probability values given their statistical interpretation and the availability of statistical data about phenomena.

With this setting, the possibility of performing an iteration process over probabilities can be considered. In this case, the iterative process of probability update will be the following:

$$P^n(A) = \begin{cases} P(A) & \text{if } n = 0, \\ P^{n-1}(A) \cdot U^n(A) & otherwise \end{cases}$$

where

$$U^i(A) = P^{i-1}(A) \cdot \prod_{\substack{\gamma \in Args \\ \langle \gamma, A \rangle \in Att}} 1 - \alpha \cdot P^{i-1}(\gamma).$$

However, we are currently interested in relative proportions among probabilities. The iterative process keeps proportions without adding further information, and it is not trivial to design a stopping criterion either. For these reasons, we only make use of a single update step in this work. The proposed formula resembles one of those proposed by Gabbay et al. [11]. However, in that work, the starting probability of the argument was not taken into account. The consequence is that the "belief" is only given by the attacks, limiting the epistemic perspective. Moreover, the hyperparameter α might introduce certain degrees of confidence in attacks. With respect to the *epistemic extension* [12, 13], this extension does not fully rely only on epistemic values, but attacks come into play to review the degree of belief of arguments. For instance, attacks from arguments with probability 1 assume a much lower impact than in our case.

Given this argumentation framework, we are ready to define the *most-probable* extension. Differently from other semantics, we define this semantics starting from an argument we call "target". We need first to specify how we consider the probability of a set. Given $\{a_1, a_2, ..., a_n\} \subseteq Args$, $P(\{a_1, a_2, ..., a_n\}) = \min\{P(a_1), P(a_2), ..., P(a_n)\}$.

Definition 13 Let $\langle Args,\ Att,\ PArgs \rangle$ G a *Simplified Probabilistic Argumentation Framework* and $t \in Args$, a set of arguments $S \subseteq Args$ is a *most-probable* extension for t, indicated as $S \in \text{most-prob}_t(G)$ iff $t \in S$, S is admissible and $\nexists Y \subset S$ such that:

- $t \in Y$

- Y is admissible
- $P(Y \setminus \{t\}) > P(S \setminus \{t\})$.

and $\nexists Z \subseteq Args$ such that:

- $S \subset Z$
- Z is admissible
- $P(Z \setminus \{t\}) = P(S \setminus \{t\})$.

In other words, S is *most-probable* with respect to t iff it is admissible, includes t, including any other argument will make the set non-admissible or reduce the minimum probability of the set, and removing any argument will not increase the minimum probability while keeping admissibility. By construction, the target argument t always belongs to most-probable sets, and its probability is not evaluated when computing the minimum, saving us from a *paradox of certainty*. Suppose $P(t) \simeq 1$, then it is likely that the singleton $\{t\}$ is the only most-probable set since adding any other arguments would probably reduce the minimum probability.

Lemma 1 There is not a unique solution for the most-probable extension.

Proof by construction, suppose G a SPAF with $Args = \{a_1, a_2, ..., a_n, s_0, s_1, t\}$ where $\{a_1, a_2, ..., a_n, t\}$ are not attacked by anyone, and s_0, s_1 attack each other. Then, both $\{a_1, a_2, ..., a_n, s_0, t\}$ and $\{a_1, a_2, ..., a_n, s_1, t\}$ are most-$\text{prob}_t(G)$.
A variant of this semantics can be easily considered if, instead of getting the most probable admissible set, we are interested only in admissible sets showing at least a probability of δ.

Definition 14 Let $\langle Args, Att, PArgs \rangle$ G a *Simplified Probabilistic Argumentation Framework*, $t \in Args$ and $\delta \in \mathbb{R}_{>}$, a set of arguments $S \subseteq Args$ is a δ*-most-probable* extension with respect to t iff S is *most-probable* with respect to t and $P(S \setminus \{t\}) >= \delta$ Note that $\delta = 0$ would reduce *threshold most-probable* to *most-probable*.

In most cases, the variant version may be more advantageous in order to not only optimise as much as possible the likelihood of extracted arguments but also to accept (resp. restrict) the result set. This extension should be used as a further step after *most-probable*. Given the probability of one of the *most-probable*, any threshold above would result in having the set, while a lower threshold provides additional information on other sets.

We report in Figure 8 a more complex example of SPAF (after probability updates) in which green nodes represent one of its *most-probable* extensions with respect to D.

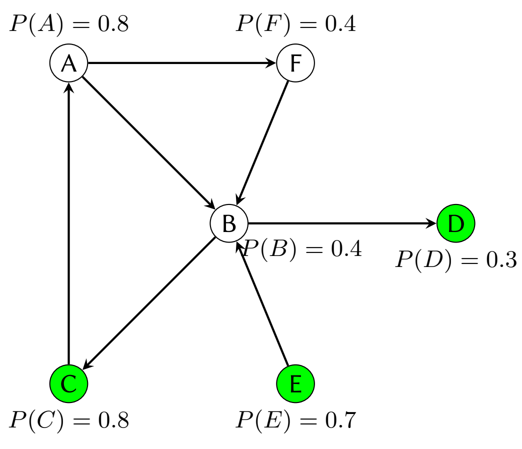

Figure 8: Most-probable extension with respect to D of a SPAF.

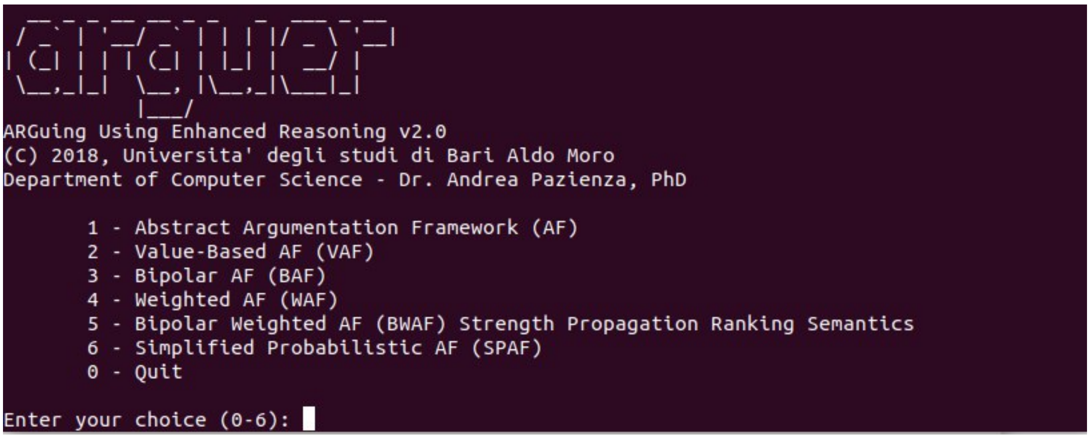

Figure 9: Main dialogue of Arguer.

Note that F belongs to an admissible set in which there is also D but it reduced minimum probability. $\{C, D, E\}$ is (with respect to D) most-probable and 0.6-most-probable, but not 0.8-most-probable.

The *most-probable* extension is integrated in a more general framework of argumentation proposed in [14, 15]. The general framework comprises a set of extensions compared to the Dung model like probabilities of arguments, attacks, supports, and weights on both attacks and supports.

4. Implementation

For implementation, we developed this probabilistic argumentation framework in a platform for argument reasoning called *ARGuing Using Enhanced Reasoning (Arguer)*. It is an interactive Prolog system for argument reasoning in which the most common frameworks and semantics have been developed. The system makes use of the YAP[1] compiler. Experiments of the proposed work have been performed on an Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz-1.50 GHz processor with 16GB of RAM.

4.1. Arguer

Arguer is a generic argumentation reasoning platform tailored for domain experts who can provide arguments in the form of propositional logic. The formalism to express arguments, attacks, or other components (e.g. weights) is predetermined. The knowledge about arguments and attacks can be fetched from a Prolog-like text file or by constructing facts directly from the interface. Based on the selected argumentation framework, several extensions can be computed.
Arguer provides the following argumentation frameworks:

- Abstract Argumentation Framework (AF)
- Value-Based AF
- Bipolar AF
- Weighted AF
- Bipolar Weighted AF
- Simplified Probabilistic AF

Figure 9 shows the main dialogue of Arguer.
Now we describe, for each framework, the formalism and the possible operations.

[1]https://www.dcc.fc.up.pt/ vsc/yap/

```
Argumentation Framework:
[[a,b,c,d,e],[[a,b],[c,b],[c,d],[d,c],[d,e],[e,e]]]

        Extension-Based Semantics

       1 - Admissible Sets
       2 - Complete Extensions
       3 - Grounded Extension
       4 - Preferred Extensions
       5 - Stable Extensions
       6 - Semi-Stable Extensions
       7 - Ideal Extension
       8 - Eager Extension
       9 - Stage Extensions
       10 - Naive Extensions
       0 - Quit
```

Figure 10: Main dialogue of Arguer.

```
Admissible Sets:
[[a,c],[a,d],[a],[c],[d],[]]
Runtime : 0.00000 sec.

Argumentation Framework:
[[a,b,c,d,e],[[a,b],[c,b],[c,d],[d,c],[d,e],[e,e]]]

        Extension-Based Semantics

       1 - Admissible Sets
       2 - Complete Extensions
       3 - Grounded Extension
       4 - Preferred Extensions
       5 - Stable Extensions
       6 - Semi-Stable Extensions
       7 - Ideal Extension
       8 - Eager Extension
       9 - Stage Extensions
       10 - Naive Extensions
       0 - Quit
```

Figure 11: Main dialogue of Arguer.

Abstract Argumentation Framework (AF). To express arguments and attacks, the predicates `argument/1` and `attack/2` are used. The first one specifies the name of the argument, and the second the relationship between the attacker (the first parameter) and the attacked (the second parameter). After loading a Prolog source made up of these facts, the following semantics can be computed:

- admissible
- complete
- grounded
- preferred
- stable
- semi-stable
- eager
- stage
- naive.

Figure 10 shows the menu after loading an abstract AF knowledge source.
Figure 11 shows the admissible sets for the previously-loaded AF.
We report in Listing 1 the Prolog code for extracting admissible (and conflict-free) sets from arguments.

Value-Based AF. It is an extension of an abstract AF in which arguments are associated with values, and values play a crucial role, as well as attacks, in choosing acceptable sets of arguments. Apart from

the predicates above, we can find `value/1` to express possible values and the `val_mapping/2` term to express that the argument in the first position is associated with the value in the second.
For this framework, the admissible, preferred and stable semantics can be computed.

Bipolar AF. This extension provides the `support/2` term to express the opposite of the attack relationship.
For this framework, the admissible, preferred and stable semantics can be computed.

Weighted AF. In this extension, for each argument, we can specify its authority in a domain. This is supported by the terms `domain/1` to specify domains and `authority/3` in which the first item is the argument, the second a value of authority, and the third the domain.

Bipolar Weighted AF. In this extension, we have the same terms as the ones available in Bipolar AF, with the introduction of `rel_weight/3` expressing attacks or support based on the positivity of the weight on relationships. The range is $[-1, 0[\cup]0, 1]$.
For this framework, the strength-propagation ranking semantics can be computed.

Simplified Probabilistic AF. In the proposed extension, the only added term is `likelihood/2` expressing the probability of each argument, and having range $]0, 1]$.
For this framework, the proposed max-probable semantics can be computed.

Listing 1: Prolog Code for Conflict-Free Sets

```
abs_arg(Argument) :-
    args:argument(Argument).
get_abstract_arguments(Arguments) :-
    findall(Argument, abs_arg(Argument), Arguments).
not_in_conflict([]) :-
    !.
not_in_conflict([Argument]) :-
    abs_arg(Argument),
    \+ attack_rel(Argument, Argument),
    !.
set_conflict_free_sets(Arguments) :-
    forall(subsets(Arguments, Args),
        (not_in_conflict(Args) ->
            assert(cf:cfree(Args)); true
        )
    ).
set_admissible_sets :-
    get_abstract_arguments(Arguments),
    set_conflict_free_sets(Arguments),
    get_conflict_free_sets(CFSets),
    set_acceptable_arguments(Arguments, CFSets).
get_conflict_free_sets(CFSets) :-
    findall(CFArgs, cf:cfree(CFArgs), CFSets).
set_acceptable_arguments(Arguments, CFSets) :-
    forall((member(CFSet, CFSets)),
        (acceptable_set(Arguments, CFSet) ->
            assertz(extensions:admissible(CFSet)); true
        )
    ).
```

4.2. Most-probable

Most-probable extension for a target argument is computed starting from the admissible sets, filtering those containing the target argument, and then an optimization function selects the arguments to be kept. Listing 2 shows the solution for probability updating after attacks, and how to normalize results for Answer Set Programming. ASP cannot manage floating numbers. For this reason, updated probabilities are normalized in the range $]0 - 100]$.

Listing 2: Prolog Code for Updating Likelihoods

```
update(Arguments, Attacks, Likelihoods) :-
    adjust_likelihoods(Arguments, Attacks, Likelihoods),
    normalize_probabilities.
adjust_likelihoods([], _, _).
adjust_likelihoods([Arg | Rest], Attacks, Likelihoods) :-
    adjust_likelihood(Arg, Attacks, Likelihoods),
    adjust_likelihoods(Rest, Attacks, Likelihoods).
adjust_likelihood(Arg, Attacks, Likelihoods) :-
    findall(X, lists:member([X,Arg], Attacks), Attackers),
    findall(X, (lists:member(A, Attackers), lists:member([A,X], Likelihoods)), ProbabilitiesAt
    Alfa is 0.2,
    compute_product(Alfa, ProbabilitiesAttackers, Product),
    AdjustedProbability is Product * OriginalProbability,
    assertz(not_normalized_probability(Arg, AdjustedProbability)).
compute_product(Alfa, [], 1) :-
    !.
compute_product(Alfa, [P | Rest], Product) :-
    TermProduct is 1 - Alfa * P,
    compute_product(Alfa, Rest, OldProduct),
    Product is TermProduct * OldProduct.
find_max([X], X).
find_max([Number | Rest], Max) :-
    find_max(Rest, MaxRest),
    Max is max(Number, MaxRest).
normalize_probabilities :-
    findall(X, not_normalized_probability(_,X), Probabilities),
    find_max(Probabilities, Max),
    Factor is 100 / Max,
    findall((Arg, Probability), not_normalized_probability(Arg, Probability), ArgProbabilities
    assert_probabilities(ArgProbabilities, Factor).
assert_probabilities([], Factor) :-
    !.
assert_probabilities([ArgProbability | Rest], Factor) :-
    ArgProbability = (Argument, Probability),
    NormalizedProbability is round(Factor * Probability),
    assertz(probability(Argument, NormalizedProbability)),
    assert_probabilities(Rest, Factor).
```

Finally, Listing 3 shows the ASP code for computing most-prob extension. The listing shows the (normalized) example reported in Figure 8, and Listing 4 the result provided by ASP after optimization.

Listing 3: ASP Code for Most-probable Extension

```
argument(a;b;c;d;e;f).
target(d).
```

```
attack(a, b). attack(a, b).
attack(b, c). attack(b, d).
attack(c, a). attack(e, b).
attack(f, b).
likelihood(a, 80).
likelihood(b, 40).
likelihood(c, 80).
likelihood(d, 30).
likelihood(e, 70).
likelihood(f, 40).

% the target is always in the most-prob
most_prob(N) :- target(N).
%all the others are candidate
{most_prob(N) : argument(N)} :- not target(N).
%arguments must be defended (or attacks free) to be good candidates
:- most_prob(A), attack(B, A), not most_prob(C) : attack(C, B).
%take minimum likelihood
min_likelihood(Min) :- Min = #min { L : most_prob(A), likelihood(A, L), not target(A) }.
%take cardinality
size_most_prob(C) :- C = #count { A : most_prob(A) }.
%take set with the maximum minimum likelihood
#maximize { L : min_likelihood(L) }.
%add as many arguments as possible without affecting the minimum likelihood
#maximize { C : size_most_prob(C) }.
#show most_prob/1.
```

Listing 4: ASP Result for Most-probable Extension

```
max_prob(d) max_prob(c) max_prob(e)
```

5. Related Work

One of the first and most successful ways of formalizing probability dependencies is with the help of *Bayesian nets* [16]. Their main limitation remains the independence assumption, although this is shared in probabilistic modelling, and they have difficulty capturing all dependencies among factors coming into play. Bayesian nets can be generalized in one of the most complete theories on uncertainty management, coming under the name of *Dempster-Shafer Theory of Evidence* [17], in which the proposed work can be considered an instance of the theory, with the introduction of the update function and the context for argumentation mining. Then, Dubois et al. [18] described the relationship between his *Possibility Theory* and previous probabilistic approaches, especially *Fuzzy Logic* [19] by Zadeh. In the case of available ground truth, *Assumption-based Systems* [20] model uncertainty starting from *acceptable* assumptions. All these models share *non-monotonicity*. Especially in the law field, there are cases in which one solution is preferred over the other. For instance, the *innocence until proven guilty* must be preserved. For these cases, Alfano et al. [21] proposed the AF with conditional preferences. In the realm of these models, we propose a variation of the Theory of Evidence, without delving into the complexity of modelling probability for uncertain rules, as in Assumption-based models. Morveli et al. [22] introduced a belief factor used to formulate the likelihood of semantics without probability updates. Following statistical interpretation, Fazzinga et al. [23] used Monte-Carlo simulation to estimate probabilities of semantics. This approach is suitable when prior information is not guaranteed or does not have enough support. Thimm et al. [24] introduced probabilities on extensions. This solution complies with several foundations [25] in the field, which are not valid in this work. More details on the psychological and social implications of resolving conflicts are available in [26].

6. Conclusions

In this work, we proposed a new representation for probabilistic argumentation frameworks, introducing a new semantics, and providing management tools and interpretation of it. Much research can be pursued in the direction of new probability updates, statistical interpretation of arguments, and applications, but also in the direction of fully exploiting the more general framework and proposing a more comprehensive semantics for the other extensions.

Acknowledgments

Thanks to Dr. Andrea Pazienza for having contributed to the development of Arguer.
This research was partially supported by project FAIR – Future AI Research (PE00000013), spoke 6 – Symbiotic AI, under the NRRP MUR program funded by the NextGenerationEU.

References

[1] J. L. Pollock, Defeasible reasoning, Cognitive science 11 (1987) 481–518.
[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. Bondarenko, P. M. Dung, R. A. Kowalski, F. Toni, An abstract, argumentation-theoretic approach to default reasoning, Artificial intelligence 93 (1997) 63–101.
[4] H. Prakken, G. Sartor, Law and logic: A review from an argumentation perspective, Artificial intelligence 227 (2015) 214–245.
[5] D. H. Warren, L. M. Pereira, F. Pereira, Prolog-the language and its implementation compared with lisp, ACM SIGPLAN Notices 12 (1977) 109–115.
[6] V. Lifschitz, Answer set programming, volume 3, Springer Heidelberg, 2019.
[7] H. Li, N. Oren, T. J. Norman, Probabilistic argumentation frameworks, in: International workshop on theorie and applications of formal argumentation, Springer, 2011, pp. 1–16.
[8] R. Fagin, J. Y. Halpern, Uncertainty, belief, and probability 1, Computational Intelligence 7 (1991) 160–173.
[9] S. Bistarelli, F. Santini, et al., Weighted argumentation., FLAP 8 (2021) 1589–1622.
[10] V. A. Uspensky, Kolmogorov and mathematical logic, The Journal of Symbolic Logic 57 (1992) 385–412.
[11] D. M. Gabbay, Equational approach to argumentation networks, Argument & Computation 3 (2012) 87–142.
[12] A. I. Goldman, An epistemological approach to argumentation, Informal Logic 23 (2003).
[13] A. Hunter, A probabilistic approach to modelling uncertain logical arguments, International Journal of Approximate Reasoning 54 (2013) 47–81.
[14] A. Pazienza, S. Ferilli, F. Esposito, S. Bistarelli, M. Giacomin, Constructing and evaluating bipolar weighted argumentation frameworks for online debating systems., in: AI^3@ AI* IA, 2017, pp. 111–125.
[15] A. Pazienza, S. Ferilli, F. Esposito, On the gradual acceptability of arguments in bipolar weighted argumentation frameworks with degrees of trust, in: Foundations of Intelligent Systems: 23rd International Symposium, ISMIS 2017, Warsaw, Poland, June 26-29, 2017, Proceedings 23, Springer, 2017, pp. 195–204.
[16] S. L. Lauritzen, D. J. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems, Journal of the Royal Statistical Society: Series B (Methodological) 50 (1988) 157–194.

[17] G. Shafer, A mathematical theory of evidence, volume 42, Princeton university press, 1976.
[18] D. Dubois, Possibility theory and statistical reasoning, Computational statistics & data analysis 51 (2006) 47–69.
[19] L. A. Zadeh, Fuzzy logic, Computer 21 (1988) 83–93.
[20] R. Haenni, Modeling uncertainty with propositional assumption-based systems, in: Applications of uncertainty formalisms, Springer, 2001, pp. 446–470.
[21] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, Abstract argumentation framework with conditional preferences, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, 2023, pp. 6218–6227.
[22] M. Morveli-Espinoza, J. C. Nieves, C. A. Tacla, Towards an imprecise probability approach for abstract argumentation 1, in: ECAI 2020, IOS Press, 2020, pp. 2921–2922.
[23] B. Fazzinga, S. Flesca, F. Parisi, On the complexity of probabilistic abstract argumentation frameworks, ACM Transactions on Computational Logic (TOCL) 16 (2015) 1–39.
[24] M. Thimm, P. Baroni, M. Giacomin, P. Vicig, Probabilities on extensions in abstract argumentation, in: Theory and Applications of Formal Argumentation: 4th International Workshop, TAFA 2017, Melbourne, VIC, Australia, August 19-20, 2017, Revised Selected Papers 4, Springer, 2018, pp. 102–119.
[25] A. Hunter, M. Thimm, Probabilistic argumentation with incomplete information, in: ECAI 2014, IOS Press, 2014, pp. 1033–1034.
[26] L. Kriesberg, Constructive conflicts: From escalation to resolution, Rowman & Littlefield, 2007.