arXiv:1608.00302v2 [cs.AI] 28 Nov 2016

Formulating Semantics of Probabilistic Argumentation by Characterizing Subgraphs: Theory and Empirical Results☆

Beishui Liao[a,b,c], Kang Xu[a,b], Huaxin Huang[a,b]

[a] *Center for the Study of Language and Cognition, Zhejiang University, China*
[b] *Institute of Logic and Cognition, Zhejiang University, China*
[c] *University of Luxembourg, Luxembourg*

Abstract

The existing approaches to formulate the semantics of probabilistic argumentation are based on the notion of possible world. Given a probabilistic argument graph (PrAG) with n nodes, up to 2^n subgraphs are blindly constructed and their extensions under a given semantics are computed. Then, the probability of a set of arguments E being an extension under a given semantics σ (denoted as $p(E^\sigma)$) is equal to the sum of the probabilities of all subgraphs each of which has the extension E. Since many irrelevant subgraphs are constructed, and in many cases, computing extensions of subgraphs is computationally intractable, these approaches are fundamentally inefficient or infeasible. In existing literature, while approximate approaches based on the Monte-Carlo simulation technique have been proposed to estimate the probability of extensions, how to improve the efficiency of computation without using the simulation technique is still an open problem. In this paper, we address this problem from the following two perspectives. First, conceptually, we define specific properties to characterize the subgraphs of a PrAG with respect to a given extension, such that the probability of a set of arguments E being an extension can be defined in terms of these properties, without (or with less) construction of subgraphs. Second, computationally, we take preferred semantics as an example, and develop algorithms to evaluate the efficiency of our approach. The results show that our approach not

Email addresses: baiseliao@zju.edu.cn (Beishui Liao), xukanguuu@163.com (Kang Xu), rw211@zju.edu.cn (Huaxin Huang)

only dramatically decreases the time for computing $p(E^\sigma)$, but also has an attractive property, which is contrary to that of existing approaches: the denser the edges of a PrAG are or the bigger the size of a given extension E is, the more efficient our approach computes $p(E^\sigma)$. Meanwhile, it is shown that under complete and preferred semantics, the problems of determining $p(E^\sigma)$ are fixed-parameter tractable.

Keywords: Probabilistic Argumentation, Computational Complexity, Computational Efficiency, Characterized Subgraphs, Fixed-Parameter Tractability

1. Introduction

In the past two decades, argumentation has been a very active research area in the field of knowledge representation and reasoning, as a nonmonotonic formalism to handle inconsistent and incomplete information by means of constructing, comparing and evaluating arguments. In 1995, Dung proposed a notion of abstract argumentation framework [1], which can be viewed as a directed graph (called *argument graph*, or defeat graph) $G = (A, R)$, in which A is a set of arguments and $R \subseteq A \times A$ is a set of attacks. Given an argument graph, specific evaluation criteria are defined to determine which arguments can be regarded as justified or acceptable. A set of arguments acceptable together is often called an extension, and the evaluation criteria or sets of extensions of an argument graph are called argumentation semantics. Dung's abstract argumentation framework and argumentation semantics lay a concrete foundation for the development of various argument systems.

However, in classical argumentation theory, the uncertainty of arguments and/or attacks is not considered. So, it could be regarded as a purely qualitative formalism. But, in the real world, arguments and/or attacks are often uncertain. So, in recent years, the importance of combining argumentation and uncertainty has been well recognized, and probability-based argumentation is gaining momentum [2, 3, 4, 5, 6]. In a *probabilistic argument graph* (or PrAG in brief), each argument is assigned with a probability, denoting the likelihood of the argument appearing in the graph[1].

[1] A probabilistic argument graph can be defined by assigning probabilities to arguments [2, 3, 7], or attacks [6], or both arguments and attacks [4]. For simplicity, in this paper we only consider the probabilistic argument graph in which only arguments are associated

Similar to classical argumentation theory, given a PrAG, a basic problem is to define the status of arguments. The existing approaches are based on the notion of possible worlds [2, 3, 4, 7]. Given a PrAG with n nodes, up to 2^n subgraphs are blindly constructed. Each subgraph corresponds to a possible world where some arguments appear while other arguments do not appear. The extensions of each subgraph are computed according to classical argumentation semantics. Then, the probability of a set of arguments E being an extension under a given semantics σ (denoted as $p(E^\sigma)$) is equal to the sum of the probabilities of all subgraphs each of which has the extension E. Since many irrelevant subgraphs are constructed, and in many cases, computing extensions of subgraphs is computationally intractable, these approaches are fundamentally inefficient or infeasible. In existing literature, while approximate approaches based on the Monte-Carlo simulation technique have been proposed to estimate the probability of extensions [4, 8], how to improve the efficiency of computation without using simulation technique is still an open problem.

Since the complexity of computing $p(E^\sigma)$ by the existing approaches is mainly caused by blindly constructing subgraphs and computing extensions of each subgraph, an intuitive question arises:

Intuitive question Is it possible to compute $p(E^\sigma)$ without (or with less) construction and computation of subgraphs?

This question has been partially answered by Fazzinga et al [9]. When analyzing the complexity of probabilistic abstract argumentation, they provided a lemma to prove that under admissible and stable semantics, the problem of computing $p(E^\sigma)$ is tractable. In this lemma, $p(E^\sigma)$ is determined by evaluating an expression which only involves the probabilities of the arguments and defeats (attacks) of a probabilistic argument graph[2]. So, in these cases, no subgraphs are constructed and computed. However, under other semantics (including complete, grounded, preferred and ideal), they only stated that the problem of computing $p(E^\sigma)$ is $FP^{\sharp P}$-complete, without further work on how the above idea can be exploited to improve the efficiency of computation under these semantics.

with probabilities.

[2]In [9], probabilities are assinged to both arguments and attacks.

Motivated by the intuitive question and the state of the art of computation of probabilistic argumentation, the research problems of the present paper are as follows.

Research problem 1 Under various argumentation semantics (including not only admissible and stable, but also complete, grounded and preferred, etc.), how to define properties to characterize the subgraphs of a PrAG with respect to an extension E, such that $p(E^\sigma)$ can be computed by using these properties, rather than by blindly constructing and computing all subgraphs of the PrAG?

Research problem 2 How to evaluate the efficiency of the new approach?

With these two research problems in mind, the rest of this paper is organized as follows. In Section 2, some notions of abstract argumentation and probabilistic abstract argumentation are reviewed to make this paper self-contained. In Sections 3 and 4, to address the first research problem, we define properties to characterize subgraphs (with respect to an extension) under different semantics (admissible, complete, grounded, preferred, and stable), and specify how the probability of a conflict-free set E being an extension can be computed by using these properties. In Section 5, to address the second research problem, algorithms are developed to evaluate the performance of the new approach (with a comparison to an existing possible worlds based approach). In Section 6, some computational properties of the new approach are briefly discussed. In Section 7, some existing work closely related to this paper is introduced and discussed. Finally, in Section 8, we conclude the paper and point out some future work.

This paper is a substantial extension of the paper introduced in [10]. The extension mainly consists of the following aspects:

- We reformulate the approach of characterizing subgraphs of a PrAG with respect to a given extension, with a more detailed analysis of the properties used to characterize subgraphs;
- further study the semantics of probabilistic argumentation by directly using properties for characterizing subgraphs;
- develop algorithms to evaluate the efficiency of the new approach; and
- analyze the computational properties of the new approach from the perspective parameterized complexity theory.

2. Preliminaries

2.1. Classical abstract argumentation

The notions of (classical) abstract argumentation were originally introduced in [1] and then extended by many researchers (please refer to [11] for an excellent introduction), including abstract argumentation framework (called *argument graph*, or *classical argument graph*, in this paper), extension-based semantics and labelling-based semantics.

An argument graph is a directed graph $G = (A, R)$, in which A is a set of nodes representing arguments and R is a set of edges representing attacks between the arguments.

Definition 1. *An argument graph is a tuple $G = (A, R)$, where A is a set of nodes representing arguments, and $R \subseteq A \times A$ is a set of edges representing attacks.*

As usual, we say that $\alpha \in A$ attacks $\beta \in A$ if and only if $(\alpha, \beta) \in R$. If $E \subseteq A$ and $\alpha \in A$ then we say that α attacks E if and only if there exists $\beta \in E$ such that α attacks β, that E attacks α if and only if there exists $\beta \in E$ such that β attacks α, and that E attacks E' if and only if there exist $\beta \in E$ and $\alpha \in E'$ such that β attacks α. Given $G = (A, R)$, for $\alpha \in A$ we write α^-_G for $\{\beta \mid (\beta, \alpha) \in R\}$; for $E \subseteq A$ we write E^-_G for $\{\beta \mid \exists \alpha \in E : (\beta, \alpha) \in R\}$ and E^+_G for $\{\beta \mid \exists \alpha \in E : (\alpha, \beta) \in R\}$. Formally, we have the following formulas.

$$\alpha^-_G = \{\beta \mid (\beta, \alpha) \in R\} \quad (1)$$

$$E^-_G = \{\beta \mid \exists \alpha \in E : (\beta, \alpha) \in R\} \quad (2)$$

$$E^+_G = \{\beta \mid \exists \alpha \in E : (\alpha, \beta) \in R\} \quad (3)$$

If without confusion, we write α^-, E^- and E^+ for α^-_G, E^-_G and E^+_G respectively.

Given an argument graph, according to certain evaluation criteria, sets of arguments (called *extensions*) are identified as acceptable together. Two important notions for the definitions of various kinds of extensions are *conflict-freeness* and *acceptability* of arguments.

Definition 2. *Let $G = (A, R)$ be an argument graph, and $E \subseteq A$ be a set of arguments.*

- *E is* conflict-free *if and only if $\nexists \alpha, \beta \in E$, such that $(\alpha, \beta) \in R$.*

- *An argument $\alpha \in A$ is acceptable with respect to (defended by) E, if and only if $\forall(\beta, \alpha) \in R$, $\exists\gamma \in E$, such that $(\gamma, \beta) \in R$.*

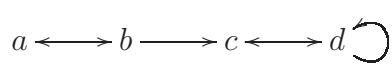
Based on the above two notions, several classes of (classical) extensions can be defined as follows.

Definition 3. *Let $G = (A, R)$ be an argument graph, and $E \subseteq A$ a set of arguments.*

- *E is* admissible *if and only if E is conflict-free, and each argument in E is acceptable with respect to E.*
- *E is preferred if and only if E is a maximal (with respect to set-inclusion) admissible set.*
- *E is complete if and only if E is admissible, and each argument that is acceptable with respect to E is in E.*
- *E is grounded if and only if E is the minimal (with respect to set-inclusion) complete extension.*
- *E is stable if and only if E is conflict-free, and each argument in $A \backslash E$ is attacked by E.*

In this paper, for convenience, we use $\sigma \in \{ad, co, pr, gr, st\}$ to represent a semantics (admissible, complete, preferred, grounded or stable). An extension under semantics σ is called a σ-extension. The set of σ-extensions of G is denoted as $\mathcal{E}_\sigma(G)$. In $G = (A, R)$, if $A = R = \varnothing$, then $\mathcal{E}_\sigma(G) = \{\varnothing\}$.

Example 1. *Let $G_1 = (A_1, R_1)$ be an argument graph illustrated as follows.*



According to Definition 3, G_1 has four admissible sets: $\varnothing$, $\{a\}$, $\{b\}$ and $\{a, c\}$, in which $\varnothing$, $\{b\}$ and $\{a, c\}$ are complete extensions, $\{b\}$ and $\{a, c\}$ are preferred extensions, $\{a, c\}$ is the only stable extension, $\varnothing$ is the unique grounded extension.

Corresponding to the extension-based approach introduced above, the *labelling-based approach* is another way to formulate argumentation semantics. Since we will use labelling-based approach to develop algorithms in

Section 5.1, some basic notions of this approach are briefly introduced here. The idea underlying the labelling-based approach is to give each argument a label, which is defined in advance. In existing literature, the set of labels is usually defined as: IN, OUT and UNDEC. The label IN indicates that the argument is explicitly accepted, the label OUT indicates that the argument is explicitly rejected, and the label UNDEC indicates that the status of the argument is undecided, meaning that one abstains from an opinion on whether the argument is accepted or rejected. Meanwhile, there could be some other choices for the set of labels. For instance, in [12], a four-valued labelling is considered. In this paper, we choose the three-valued-labelling, which can be formally defined as follows.

Definition 4 (Labelling). *Given an argument graph $G = (A, R)$ and three labels* IN*,* OUT *and* UNDEC*, a labelling is a total function:*

$$\mathcal{L} : A \mapsto \{\mathrm{IN}, \mathrm{OUT}, \mathrm{UNDEC}\} \tag{4}$$

Let $in(\mathcal{L}) = \{\alpha \mid \mathcal{L}(\alpha) = \mathrm{IN}\}$, $out(\mathcal{L}) = \{\alpha \mid \mathcal{L}(\alpha) = \mathrm{OUT}\}$, and $undec(\mathcal{L}) = \{\alpha \mid \mathcal{L}(\alpha) = \mathrm{UNDEC}\}$. A labelling $\mathcal{L}$ is often represented as a triple of the form $(in(\mathcal{L}), out(\mathcal{L}), undec(\mathcal{L}))$.

One of criteria for labeling-based semantics is whether a label assigned to an argument is legal. According to Definition 4, given a labeling $\mathcal{L}$, the status assigned to each argument might not be legal. We say that assigning IN to an argument is legal if and only if all its attackers have been assigned OUT; assigning OUT to an argument is legal if and only if one of its attackers has been assigned IN; and assigning UNDEC to an argument is legal if and only if not all its attacks are labeled OUT and it does not have an attacker that is labeled IN. Formally, we have the following definition.

Definition 5 (Legal labelling). *Let $\mathcal{L}$ be a labeling of an argument graph $G = (A, R)$ and $\alpha \in A$.*

- *α is legally* IN *if and only if $\mathcal{L}(\alpha) = \mathrm{IN}$ and for all $\beta \in A$, if $(\beta, \alpha) \in R$, then $\mathcal{L}(\beta) = \mathrm{OUT}$.*
- *α is legally* OUT *if and only if $\mathcal{L}(\alpha) = \mathrm{OUT}$ and there exists $\beta \in A$, such that $(\beta, \alpha) \in R$, and $\mathcal{L}(\beta) = \mathrm{IN}$.*
- *α is legally* UNDEC *if and only if $\mathcal{L}(\alpha) = \mathrm{UNDEC}$ and*
  (1) there exists $\beta \in A$, such that $(\beta, \alpha) \in R$, and $\mathcal{L}(\beta) \neq \mathrm{OUT}$, and
  (2) for all $\beta \in A$, if $(\beta, \alpha) \in R$, then $\mathcal{L}(\beta) \neq \mathrm{IN}$.

According to the notion of legal labelling, the notion of illegal labelling can be defined as follows.

Definition 6 (Illegal labelling). *Let $\mathcal{L}$ be a labeling of an argument graph $G = (A, R)$ and $\alpha \in A$.*

- *α is illegally* IN *if and only if $\mathcal{L}(\alpha) =$* IN*, but α is not legally* IN*.*
- *α is illegally* OUT *if and only if $\mathcal{L}(\alpha) =$* OUT*, but α is not legally* OUT*.*
- *α is illegally* UNDEC *if and only if $\mathcal{L}(\alpha) =$* UNDEC*, but α is not legally* UNDEC*.*

Based on the notions of legal labelling, labeling-based semantics can be defined as follows.

Definition 7 (Labeling-Based Semantics). *Let $\mathcal{L}$ be a labeling of an argument graph $G = (A, R)$.*

- *$\mathcal{L}$ is an admissible labeling, if and only if each argument that is labeled* IN *is legally* IN*, and each argument that is labeled* OUT *is legally* OUT*.*
- *$\mathcal{L}$ is a complete labeling, if and only if it is an admissible labeling, and each argument that is labeled* UNDEC *is legally* UNDEC*.*
- *$\mathcal{L}$ is a grounded labeling, if and only if it is a complete labeling, and $in(\mathcal{L})$ is minimal (with respect to set inclusion).*
- *$\mathcal{L}$ is a preferred labeling, if and only if it is a complete labeling, and $in(\mathcal{L})$ is maximal (with respect to set inclusion).*
- *$\mathcal{L}$ is a stable labeling, if and only if it is a complete labeling, and $undec(\mathcal{L}) = \varnothing$.*

Based on the above notions, Modgil and Caminada developed algorithms (called MC algorithms)[13] to compute the preferred labellings and the grounded

labelling of an argument graph[3].

The MC algorithm for computing preferred labellings is realized by computing *admissible labellings* that maximize the number of arguments that are legally IN. Here, admissible labellings are generated by starting with a labelling that labels all arguments IN and then iteratively, selects arguments that are illegally IN (or *super-illegally IN*) and applies a *transition step* to obtain a new labelling, until a labelling is reached in which no argument is illegally IN. In this algorithm, the notions of *super-illegally IN* and *transition step* are introduced as follows. For more details about the MC algorithms, please refer to [13].

First, since all arguments are initially labelled IN, some of which might be illegal. To get an admissible labelling which might be a preferred labelling, it is necessary to change the label of each argument that is illegally IN, preferably without creating any arguments that are illegally OUT. The notion of a transition step is used for this purpose. In other words, a transition step basically takes an argument that is illegally IN and relabels it to OUT. It then checks if, as a result of this, one or more arguments have become illegally OUT. If this is the case, then these arguments are relabelled to UNDEC. Formally, the notion of *transition step* is defined as follows [13].

Definition 8 (Transition step). *Let $\mathcal{L}$ be a labelling for $G = (A, R)$ and α be an argument that is illegally IN in $\mathcal{L}$. A transition step on α in $\mathcal{L}$ consists of the following:*

- *the label of α is changed from IN to OUT;*
- *for every $\beta \in \{\alpha\} \cup \{\gamma \mid (\alpha, \gamma) \in R\}$, if β is illegally OUT, then the label of β is changed from OUT to UNDEC.*

Second, if we select arbitrarily the arguments that are illegally IN to do transition steps, then we might obtain some admissible labellings that are not

[3]It is worth to mention that in recent years, there are various approaches for computing the semantics of argumentation, including reduction approaches (e.g. the alrogithms based on ASP slovers) and direct approaches (e.g. the MC algorithms). Some of them have appeared in the International Competition on Computational Models of Argumentation (http://argumentationcompetition.org/2015/solvers.html). Since the choice of different implemention approaches does not basically affect the empirical results of our approach, for simplicity and without loss of generarity, we only introduce and exploit the MC algorithm for computing preferred labellings.

complete labellings (and therefore not preferred labellings). To improve the efficiency of computation, in the MC algorithm for preferred labellings, they proposed a notion, called *super-illegally* IN. It is said that an argument α in $\mathcal{L}$ that is illegally IN, is also super-illegally IN if and only if it is attacked by an argument β that is legally IN in $\mathcal{L}$, or UNDEC in $\mathcal{L}$. This notion can be used to guide the choice of arguments on which to perform transition steps, such that the non-complete labellings can be avoided.

2.2. Probabilistic abstract argumentation

The notions of probabilistic abstract argumentation are defined by combining the notions of classical abstract argumentation and those of probabilistic theory, including probabilistic argument graph and its semantics.

According to [7], we have the following definition.

Definition 9. *A probabilistic argument graph (or PrAG for short) is a triple $G^p = (A, R, p)$ where $G = (A, R)$ is an argument graph and $p : A \rightarrow [0, 1]$ is a probability function assigning to every argument $\alpha \in A$ a probability $p(\alpha)$ that α appears (and hence a probability $1 - p(\alpha)$ that α does not appear).*

In existing literature, the semantics of a PrAG is defined according to the notion of possible world. Given a PrAG, a possible world represents a scenario consisting of some subset of the arguments and attacks in the graph. So, given a PrAG with n nodes, there are up to 2^n subgraphs with nonzero probability. A subgraph induced by a set $A' \subseteq A$ is represented as $G' = (A', R')$, in which $R' = R \cap (A' \times A')$. For convenience, we also use $G_{\downarrow A'}$ to denote a subgraph $G' = (A', R')$. Under a semantics $\sigma \in \{ad, co, pr, gr, st\}$, the extensions of each subgraph are computed according to the definition of classical argumentation semantics. Then, the probability that a set of arguments $E \subseteq A$ is a σ-extension, denoted as $p(E^\sigma)$, is the sum of the probability of each subgraph for which E is a σ-extension. In calculating the probability of each subgraph, we assume independence of arguments appearing in a graph. A discussion about the assumption of independence of arguments is presented in Section 7.1.

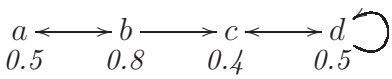
For simplicity, let us abuse the notation, using $p(\bar{\alpha})$ to denote $1 - p(\alpha)$. Then, the probability of subgraph G', denoted $p(G')$, can be defined as follows.

$$p(G') \quad = \quad (\Pi_{\alpha \in A'}\, p(\alpha)) \times (\Pi_{\alpha \in A \setminus A'}\, p(\bar{\alpha})) \tag{5}$$

Given a PrAG $G^p = (A, R, p)$, let $Q_\sigma(E)$ denote the set of subgraphs of G, each of which has an extension E under a given semantics $\sigma \in \{ad, co, pr, gr, st\}$. Based on formula (5), $p(E^\sigma)$ is defined as follows [7].

$$p(E^\sigma) \quad = \quad \Sigma_{G' \in Q_\sigma(E)}\, p(G') \tag{6}$$

Example 2. *Let $G_1^p = (A_1, R_1, p)$ be a PrAG (illustrated as follows), where $p(a) = 0.5$, $p(b) = 0.8$, $p(c) = 0.4$ and $p(d) = 0.5$.*



The subgraphs of G_1^p are presented in Table 1.

According to formula (6), there are 5 preferred extensions with non-zero probability:

$$\begin{aligned}
p(\varnothing^{pr}) &= p(G_1^{15}) + p(G_1^{16}) = 0.06 \\
p(\{a\}^{pr}) &= p(G_1^{3}) + p(G_1^{4}) + p(G_1^{7}) + p(G_1^{8}) = 0.3 \\
p(\{b\}^{pr}) &= p(G_1^{1}) + p(G_1^{2}) + p(G_1^{3}) + p(G_1^{4}) + p(G_1^{9}) + p(G_1^{10}) \\
&\quad + p(G_1^{11}) + p(G_1^{12}) = 0.8 \\
p(\{c\}^{pr}) &= p(G_1^{13}) + p(G_1^{14}) = 0.04 \\
p(\{a, c\}^{pr}) &= p(G_1^{1}) + p(G_1^{2}) + p(G_1^{5}) + p(G_1^{6}) = 0.2
\end{aligned}$$

This example shows that by using the existing possible worlds based approach, in order to obtain the probability that a set E of arguments is an extension under a given semantics (i.e., $p(E^\sigma)$), one has to compute the extensions of all subgraphs under this semantics, although some of these subgraphs have no extension E. Since many irrelevant subgraphs are constructed and computed, and in many cases, computing extensions of subgraphs is computationally intractable, this possible worlds based approach is fundamentally inefficient or infeasible. In [8], Fazzinga et al proposed a new approach and proved that under admissible and stable semantics, the problem of determining $p(E^\sigma)$ is tractable. However, under complete, grounded and preferred semantics, the problem of determining $p(E^\sigma)$ is $FP^{\sharp P}$-complete. This calls for developing more efficient approaches, including the approximate approaches introduced in [4] and [8].

	Subgraphs	Probability of subgraph	Preferred extensions
G_1^1	$a \leftrightarrow b \rightarrow c \leftrightarrow d \circlearrowleft$	0.08	$\{b\}, \{a, c\}$
G_1^2	$a \leftrightarrow b \rightarrow c$	0.08	$\{b\}, \{a, c\}$
G_1^3	$a \leftrightarrow b \quad d \circlearrowleft$	0.12	$\{a\}, \{b\}$
G_1^4	$a \leftrightarrow b$	0.12	$\{a\}, \{b\}$
G_1^5	$a \quad c \leftrightarrow d \circlearrowleft$	0.02	$\{a, c\}$
G_1^6	$a \quad c$	0.02	$\{a, c\}$
G_1^7	$a \quad d \circlearrowleft$	0.03	$\{a\}$
G_1^8	a	0.03	$\{a\}$
G_1^9	$b \rightarrow c \leftrightarrow d \circlearrowleft$	0.08	$\{b\}$
G_1^{10}	$b \rightarrow c$	0.08	$\{b\}$
G_1^{11}	$b \quad d \circlearrowleft$	0.12	$\{b\}$
G_1^{12}	b	0.12	$\{b\}$
G_1^{13}	$c \leftrightarrow d \circlearrowleft$	0.02	$\{c\}$
G_1^{14}	c	0.02	$\{c\}$
G_1^{15}	$d \circlearrowleft$	0.03	$\{\}$
G_1^{16}		0.03	$\{\}$

Table 1: Subgraphs of G_1

3. Characterized subgraphs with respect to an extension

Given a PrAG, since the probability of a set of arguments E being an extension under a given semantics σ (i.e. $p(E^\sigma)$) is equal to the sum of the probabilities of the subgraphs each of which has an extension E, the main issue is to identify the subgraphs. As mentioned in Section 1, unlike the existing approaches, we define general properties to characterize the subgraphs, such that $p(E^\sigma)$ can be computed by using these properties, rather than by blindly constructing and computing all subgraphs of the PrAG.

To begin with, let us introduce a notion of *σ-subgraph with respect to an extension*: If a subgraph has a σ-extension E, then it is called a σ-subgraph with respect to E. Formally, we have the following definition.

Definition 10. *Let $G^p = (A, R, p)$ be a PrAG, $G = (A, R)$ be the corresponding classical argument graph, $G_{\downarrow A'}$ be a subgraph of G where $A' \subseteq A$, and $E \subseteq A$ be a set of arguments. We say that $G_{\downarrow A'}$ is a σ-subgraph of G with respect to E, if and only if $G_{\downarrow A'}$ has a σ-extension E, where $\sigma \in \{ad, co, pr, gr, st\}$.*

Example 3. *Consider G_1^p in Example 2. Given $E_1 = \{a\}$, G_1^3, G_1^4, G_1^7 and G_1^8 are preferred subgraphs of G_1^p with respect to E_1.*

Then, given a PrAG $G^p = (A, R, p)$, a set of arguments $E \subseteq A$ and a semantics $\sigma \in \{ad, co, pr, gr, st\}$, a function (called *subgraph identification function*) is used to map E to a set of σ-subgraphs of G with respect to E.

Definition 11. *Let $G^p = (A, R, p)$ be a PrAG, and $G = (A, R)$ be the corresponding argument graph. Let $\mathbb{G} = \{G_{\downarrow A'} \mid A' \in 2^A\}$ be the set of all subgraphs of G. A subgraph identification function under a given semantics $\sigma \in \{ad, co, pr, gr, st\}$ (denoted as ρ^σ) is defined as a mapping:*

$$\rho^\sigma : 2^A \rightarrow 2^{\mathbb{G}} \tag{7}$$

such that given $E \in 2^A$, for all $G' \in \rho^\sigma(E)$, G' is a σ-subgraph of G with respect to E.

In Definition 11, ρ^σ ($\sigma \in \{ad, co, pr, gr, st\}$) can be understood as a class of functions (i.e., ρ^{ad}, ρ^{co}, ρ^{pr}, ρ^{gr} and ρ^{st}), each of which is a function under a given semantics.

These subgraph identification functions can be instantiated in different ways. A simple but inefficient way is to construct the set of all subgraphs of G (i.e., $\mathbb{G}$) and then for each subgraph to verify whether it has an extension E. In terms of this approach, subgraphs are constructed blindly, although many of them are irrelevant. And, for each subgraph, the algorithm to verify E being an extension of the subgraph might be intractable (e.g., under preferred semantics, the problem of verifying whether E is an extension is coNP-complete [33]).

To cope with this problem, we introduce as follows another way to instantiate the subgraph identification functions. In this new approach, properties related to E are used to characterize the set of subgraphs each of which has an extension E. Since $p(E^\sigma)$ may be computed by using these properties (please refer to Section 4 for details), the characterized subgraphs can be kept (completely or partially) implicit, rather than explicitly constructed and computed (although the subgraphs may be also explicitly represented according to the properties, as presented in formulas (8)-(12)).

Since when E is not conflict-free, the set of characterized subgraphs with respect to E is an empty set, for simplicity, when talking about the set of characterized subgraphs with respect to a set of arguments E, we only consider the cases where E is conflict-free.

Under different semantics, properties used to characterize subgraphs may vary. However, they are all based on the following components related to E (as illustrated in Figure 1)[4]:

1. E;
2. $E_G^- \backslash E_G^+$: the set of arguments each of which attacks E but is not attacked by E;
3. E_G^+: the set of arguments each of which is attacked by E;
4. $I = A \backslash (E \cup E_G^+ \cup E_G^-)$: the set of arguments each of which is not in E, E_G^+ or E_G^-. We call I the set of remaining arguments (of G with respect to E) that indirectly affects E being a σ-extension.

Firstly, under admissible semantics, each admissible subgraph can be characterized by the following two properties (as illustrated in Figure 2):

[4]The slices of the pie are used to indicate different components related to E. Their sizes are not important.

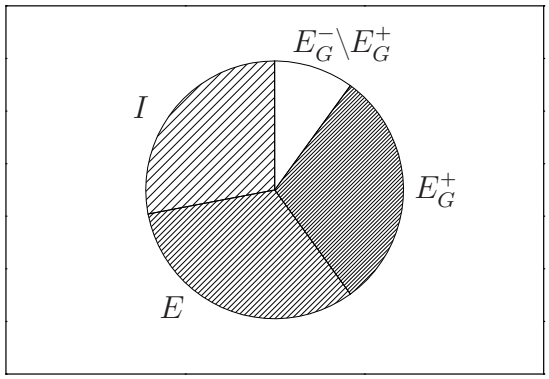


Figure 1: Four components of G w.r.t. E. It holds that $E \cup (E_G^- \backslash E_G^+) \cup E_G^+ \cup I = A$.

Prop1: All arguments in E appear in the subgraphs; and

Prop2: All arguments in $E_G^- \backslash E_G^+$ do not appear in the subgraph (while the appearance of arguments in any subset of $I \cup E_G^+$ does not affect E being an extension of the subgraph).

Prop2 means that every argument in E is acceptable with respect to E. Given that E is conflict-free and every argument in E is acceptable with respect to E, E is an admissible extension. So, by definition, the subgraph is an admissible subgraph.

According to the above analysis, we have the following proposition.

Proposition 1. *Let $G^p = (A, R, p)$ be a PrAG, $G = (A, R)$ be a corresponding argument graph, and $E \subseteq A$ be a conflict-free set of arguments. Then, for all $B \in 2^{I \cup E_G^+}$, $G_{\downarrow E \cup B}$ is an admissible subgraph of G with respect to E.*

Proof. We need to verify that E is an admissible set of $G_{\downarrow E \cup B}$. Since E is conflict-free, we only need to prove that $\forall \alpha \in E$, α is acceptable with respect to E. Since E is conflict-free and there is no interaction between I and E, it holds that α is not attacked by the arguments in $E \cup (I \cap B)$. And, $\forall \beta \in E_G^+ \cap B$, no matter whether β attacks α, by the definition of E_G^+, β is attacked by E. In summary, $\forall \beta \in E \cup (I \cap B) \cup (E_G^+ \cap B) = E \cup B$, β either does not attack α or is attacked by E. So, $\forall \alpha \in E$, α is acceptable with respect to E.

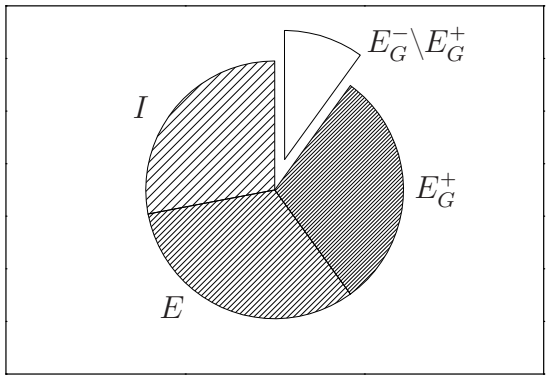


Figure 2: Given a PrAG $G = (A, R, p)$, a subgraph G' is an admissible subgraph w.r.t. E if and only if arguments in E appear, arguments in $E_G^- \backslash E_G^+$ do not appear, while arguments in any subset of $I \cup E_G^+$ may apear in the subgraph.

According to Proposition 1, the set of admissible subgraphs $\rho^{ad}(E)$ can be specified as follows:

$$\rho^{ad}(E) = \{G_{\downarrow E \cup B} \mid B \in 2^{I \cup E_G^+}\} \tag{8}$$

Example 4. *Consider G_1^p in Example 2 again. According to formula (8), there are eight admissible subgraphs with respect to $\{a\}$: G_1^1, G_1^2, ..., G_1^8 (as shown in the third column of Table 2), i.e., $\rho^{ad}(\{a\}) = \{G_1^1, G_1^2, \ldots, G_1^8\}$.*

Secondly, under stable semantics, each stable subgraph can be characterized by the following two properties (as illustrated in Figure 3):

Prop1: All arguments in E appear in the subgraph.

Prop3: All arguments in $A \backslash (E \cup E_G^+) = I \cup (E_G^- \backslash E_G^+)$ do not appear in the subgraph.

Prop3 means that for each argument α in $A \backslash E$, if it appears in the subgraph, then it is attacked by E (i.e., $\alpha \in E_G^+$). Given that E is conflict-free and for every argument that is not in E it is attacked by E, E is a stable extension. So, by definition, the subgraph is a stable subgraph.

Formally, we have the following proposition.

	subgraph	admissible subgraph w.r.t. $\{a\}$	comple subgraph w.r.t. $\{a\}$	stable subgraph w.r.t. $\{a\}$	preferred subgraph w.r.t. $\{a\}$	grounded subgraph w.r.t. $\{a\}$
G_1^1	$a \leftrightarrow b \rightarrow c \leftrightarrow d \circlearrowleft$	Yes	Yes	No	No	No
G_1^2	$a \leftrightarrow b \rightarrow c$	Yes	No	No	No	No
G_1^3	$a \leftrightarrow b \quad d \circlearrowleft$	Yes	Yes	No	Yes	No
G_1^4	$a \leftrightarrow b$	Yes	Yes	Yes	Yes	No
G_1^5	$a \quad c \leftrightarrow d \circlearrowleft$	Yes	Yes	No	No	Yes
G_1^6	$a \quad c$	Yes	No	No	No	No
G_1^7	$a \quad d \circlearrowleft$	Yes	Yes	No	Yes	Yes
G_1^8	a	Yes	Yes	Yes	Yes	Yes
G_1^9	$b \rightarrow c \leftrightarrow d \circlearrowleft$	No	No	No	No	No
G_1^{10}	$b \rightarrow c$	No	No	No	No	No
G_1^{11}	$b \quad d \circlearrowleft$	No	No	No	No	No
G_1^{12}	b	No	No	No	No	No
G_1^{13}	$c \leftrightarrow d \circlearrowleft$	No	No	No	No	No
G_1^{14}	c	No	No	No	No	No
G_1^{15}	$d \circlearrowleft$	No	No	No	No	No
G_1^{16}		No	No	No	No	No

Table 2: σ-subgraphs of G_1 with respect to $\{a\}$

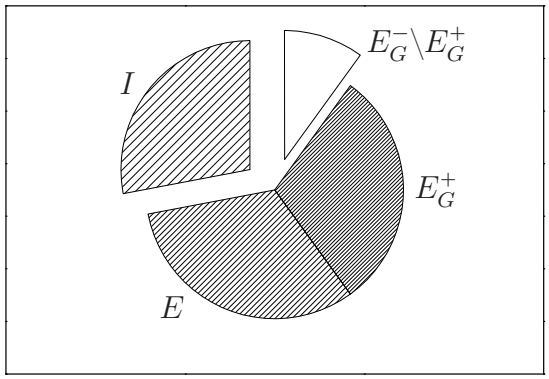


Figure 3: Given a PrAG $G = (A, R, p)$, a subgraph G' is a stable subgraph w.r.t. E if and only if arguments in E appear, arguments in $I \cup (E_G^- \backslash E_G^+)$ do not appear, while arguments in any subset of E_G^- may appear.

Proposition 2. *Let $G^p = (A, R, p)$ be a PrAG, $G = (A, R)$ be a corresponding argument graph, and $E \subseteq A$ be a conflict-free set of arguments. Then, for all $B \in 2^{E_G^+}$, $G_{\downarrow E \cup B}$ is a stable subgraph of G with respect to E.*

Proof. Since E is conflict-free, to prove E being a stable extension of $G_{\downarrow E \cup B}$, we only need to verify that $\forall \alpha \in (E \cup B) \backslash E = B$, α is attacked by E. Since $\alpha \in B \subseteq E_G^+$, by the definition of E_G^+, α is attacked by E.

According to Proposition 2, the set of stable subgraphs $\rho^{st}(E)$ can be specified as follows:

$$\rho^{st}(E) = \{G_{\downarrow E \cup B} \mid B \in 2^{E_G^+}\} \tag{9}$$

Thirdly, under other semantics (complete, grounded and preferred), the set of remaining arguments $I = A \backslash (E \cup E_G^+ \cup E_G^-)$ plays a very important role in characterizing σ-subgraphs.

Let $G_{\downarrow E \cup B}$ (where $B \in 2^{I \cup E_G^+}$) be an admissible subgraph of G with respect to E, and $B' = B \cap I$. Whether $G_{\downarrow E \cup B}$ is a complete subgraph with respect to E is determined by a property of the arguments in $B' \in 2^I$. Intuitively, if the following property holds, then $G_{\downarrow E \cup B}$ is a complete subgraph with respect to E (as illustrated in Figure 4):

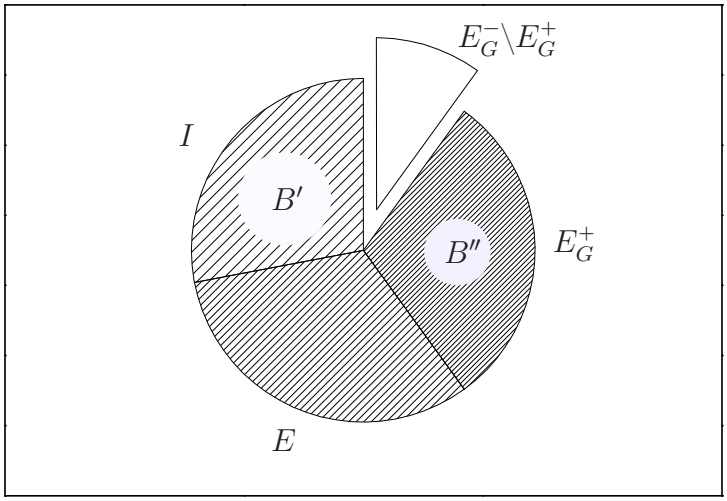


Figure 4: Given a PrAG $G = (A, R, p)$, a subgraph G' is a complete subgraph w.r.t. E if and only if it is an admissible subgraph w.r.t. E, and each argument in B' is attacked by some arguments in B'.

Prop4: For all $\alpha \in B'$, α is attacked by B'.

This property means that for every remaining argument $\alpha \in B'$, α is not acceptable with respect to E. Based on this property, we have the following proposition.

Proposition 3. *Let $G^p = (A, R, p)$ be a PrAG, $G = (A, R)$ be a corresponding argument graph, $E \subseteq A$ be a conflict-free set of arguments. For all $B \in 2^{I \cup E_G^+}$, $G_{\downarrow E \cup B}$ is a complete subgraph of G with respect to E, if and only if $\forall \alpha \in B'$, $\alpha_G^- \cap B' \neq \emptyset$.*

Proof. Since $B \in 2^{I \cup E_G^+}$, according to Proposition 1, $G_{\downarrow E \cup B}$ is an admissible subgraph.

($\Rightarrow$:) When $G_{\downarrow E \cup B}$ is a complete subgraph of G with respect to E, assume that $\exists \alpha \in B'$ such that $\alpha_G^- \cap B' = \emptyset$. It follows that α is acceptable with respect to E, and therefore E is not a complete extension, contradicting $G_{\downarrow E \cup B}$ is a complete subgraph with respect to E.

($\Leftarrow$:) For all $\alpha \in B'$, for all $\beta \in \alpha_G^- \cap B'$, β can not be attacked by the arguments in E. Otherwise, β is in E_G^+, contradicting $\beta \in B' \subseteq I$ and $I \cap E_G^+ = \emptyset$. Since $\alpha_G^- \cap B' \neq \emptyset$, α is not acceptable with respect to E. Since E is an admissible set, E is a complete extension. According to Definition 10, $G_{\downarrow E \cup B}$ is a complete subgraph of G with respect to E.

According to Proposition 3, the set of complete subgraphs $\rho^{co}(E)$ can be specified as follows:

$$\rho^{co}(E) = \{G_{\downarrow E \cup B} \mid (B \in 2^{I \cup E_G^+}) \wedge (\forall \alpha \in B' : \alpha_G^- \cap B' \neq \emptyset)\} \qquad (10)$$

Example 5. *Among the eight admissible subgraphs, except G_1^2 and G_1^6, others are complete subgraphs with respect to $\{a\}$ (as shown in the fourth column of Table 2), i.e., $\rho^{co}(\{a\}) = \{G_1^1, G_1^3, G_1^4, G_1^5, G_1^7, G_1^8\}$.*

With regard to G_1^2, $B' = \{c\}$. Since $c_{G_1}^- = \emptyset$, G_1^2 is not a complete subgraph with respect to $\{a\}$. Similarly, G_1^6 is not a complete subgraph with respect to $\{a\}$.

Then, under preferred semantics, given a complete subgraph $G_{\downarrow E \cup B}$ (where $B \in 2^{I \cup E_G^+}$), whether $G_{\downarrow E \cup B}$ is a preferred subgraph is determined by a property of the subgraph induced by $B' \in 2^I$. More specifically, if the following property holds, then $G_{\downarrow E \cup B}$ is a preferred subgraph:

Prop5: $G_{\downarrow B'}$ has only an empty admissible extension.

Proposition 4. *Let $G^p = (A, R, p)$ be a PrAG, and $E \subseteq A$ be a conflict-free set of arguments. Then, for all $B \in 2^{I \cup E_G^+}$, $G_{\downarrow E \cup B}$ is a preferred subgraph of G with respect to E if and only if $G_{\downarrow E \cup B}$ is a complete subgraph of G with respect to E, and $\mathcal{E}_{ad}(G_{\downarrow B'}) = \{\emptyset\}$.*

Proof. ($\Rightarrow$): Assume the contrary, i.e., $G_{\downarrow B'}$ has a non-empty admissible extension $E' \subseteq B'$. It follows that $E \cup E'$ is admissible, in that:

- $E \cup E'$ is conflict-free: both E and E' are conflict-free; E does not attack E' (otherwise, $E' \cap E_G^+ \neq \emptyset$, contradicting $E' \subseteq B'$); E' does not attack E (otherwise, E attacks E', contradiction).
- $\forall \alpha \in E'$, α is acceptable with respect to $E \cup E'$.

So, $E \cup E'$ is an admissible extension of $G_{\downarrow E \cup B}$. So, E is not a preferred extension of $G_{\downarrow E \cup B}$, contradicting "$G_{\downarrow E \cup B}$ is a preferred subgraph of G with respect to E".

($\Leftarrow$): Since $G_{\downarrow B'}$ has only one empty admissible extension, no argument in B' is acceptable with respect to E or any conflict-free superset of E. It turns out that E is a preferred extension of $G_{\downarrow E \cup B}$, i.e., $G_{\downarrow E \cup B}$ is a preferred subgraph of G with respect to E.

According to Proposition 4, the set of preferred subgraphs $\rho^{pr}(E)$ can be specified as follows:

$$\rho^{pr}(E) = \{G_{\downarrow E \cup B} \in \rho^{co}(E) \mid \mathcal{E}_{ad}(G_{\downarrow B'}) = \{\varnothing\}\} \tag{11}$$

Example 6. *Continue Example 5. Among the six complete subgraphs, except G_1^1 and G_1^5, others are preferred subgraphs with respect to $\{a\}$ (as shown in the sixth column of Table 2).*

With regard to G_1^1, $B' = \{c, d\}$. Then, $\mathcal{E}_{ad}(G_{\downarrow B'}) = \{\{c\}\} \neq \{\varnothing\}$. So, G_1^1 is not a preferred subgraph with respect to $\{a\}$. Similarly, G_1^5 is not a preferred subgraph with respect to $\{a\}$.

Finally, given a complete subgraph $G_{\downarrow E \cup B}$, let $B'' = B \cap E_G^+$. In order to verify whether it is a grounded subgraph, we may simply check whether $G_{\downarrow E \cup B''}$ has a grounded extension E. To simplify the computation, we may divide B'' in to two disjoint subsets B_1'' and B_2'', where $B_1'' = B \cap (E_G^+ \backslash E_G^-)$ and $B_2'' = B \cap (E_G^+ \cap E_G^-)$, as illustrated in Figure 5. Note that arguments in $E_G^+ \backslash E_G^-$ do not affect the the status of arguments in E. So, if the following property holds, then $G_{\downarrow E \cup B}$ is a grounded subgraph:

Prop6: $G_{\downarrow E \cup B_2''}$ has a grounded extension E.

Proposition 5. *Let $G^p = (A, R, p)$ be a PrAG, $E \subseteq A$ be a conflict-free set of arguments. For all $B \in 2^{I \cup E_G^+}$, $G_{\downarrow E \cup B}$ is a grounded subgraph of G with respect to E if and only if $G_{\downarrow E \cup B}$ is a complete subgraph of G with respect to E, and E is a grounded extension of $G_{\downarrow E \cup B_2''}$.*

Proof. ($\Rightarrow$): Since $G_{\downarrow E \cup B}$ is a grounded subgraph of G with respect to E, it holds that E is the grounded extension of $G_{\downarrow E \cup B}$. First, since a grounded extension is also a complete extension, E is a complete extension of $G_{\downarrow E \cup B}$,

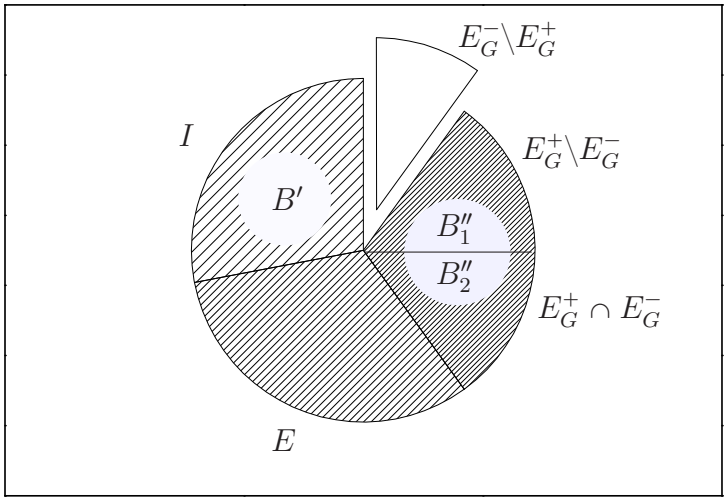


Figure 5: Given a PrAG $G = (A, R, p)$, a subgraph G' is a grounded subgraph w.r.t. E if and only if it is a complete subgraph w.r.t. E, and $G_{\downarrow E \cup B_2''}$ has a grounded extension E.

i.e., $G_{\downarrow E \cup B}$ is a complete subgraph of G with respect to E. Second, given that E is the grounded extension of $G_{\downarrow E \cup B}$, since E does not receive any attacks from B' and B_1'', according to the directionality of grounded semantics [14], it holds that E is the grounded extension of $G_{\downarrow E \cup B_2''}$, where $B_2'' = B \backslash (B' \cup B_1'')$.

($\Leftarrow$): Since $G_{\downarrow E \cup B}$ is a complete subgraph of G with respect to E, it holds that E is a complete extension of $G_{\downarrow E \cup B}$. Now, we need to verify that E is a minimal complete extension of $G_{\downarrow E \cup B}$. Assume the contrary. There exists $E' \subset E$ such that E' is a grounded extension of $G_{\downarrow E \cup B}$. According to the previous proof, it turns out that E' is a grounded extension of $G_{\downarrow E \cup B_2''}$, contradicting E is a grounded extension of $G_{\downarrow E \cup B_2''}$.

According to Proposition 5, the set of grounded subgraphs $\rho^{gr}(E)$ can be specified as follows:

$$\rho^{gr}(E) = \{G_{\downarrow E \cup B} \in \rho^{co}(E) \mid \mathcal{E}_{gr}(G_{\downarrow E \cup B_2''}) = \{E\}\} \tag{12}$$

Example 7. *Continue Example 5. Among the six complete subgraphs, G_1^5 and G_1^7 and G_1^8 are grounded subgraphs with respect to $\{a\}$ (as shown in the last column of Table 2).*

4. Semantics of probabilistic argumentation

According to the theory introduced in the previous section, given a PrAG $G^p = (A, R, p)$, a conflict-free set of arguments $E \subseteq A$ and a semantics $\sigma \in \{ad, co, pr, gr, st\}$, a set of σ-subgraphs with respect to E can be specified in terms of different properties related to E. Given **Prop1** - **Prop6** and formulas (8) - (12), we may define semantics of probabilistic argumentation by the following two approaches.

In the first place, according to formulas (8) - (12) and (6), semantics of probabilistic argumentation, i.e., the probability of E being a σ-extension (denoted as $p(E^\sigma)$), can be directly represented as follows.

$$p(E^\sigma) \quad = \quad \Sigma_{G' \in \rho^\sigma(E)} \, p(G') \tag{13}$$

Note that $Q_\sigma(E)$ in formula (6) is replaced by $\rho^\sigma(E)$ in formula (13).

In this approach, although the characterized subgraphs are explicitly represented, they are not constructed blindly, but defined according to specific properties. Therefore, the construction of most irrelevant subgraphs is avoided. Meanwhile, under admissible, complete and stable semantics, no computation of extensions is needed, while under preferred and grounded semantics, only the extension of the subgraphs induced by B' (resp. $E \cup B''$) is needed. Note that the number (resp. the average size) of the subgraphs induced by B' (resp. $E \cup B''$) is usually much smaller than that of the subgraphs induced by $A' \in 2^A$.

In the second place, a more efficient approach to define semantics of probabilistic argumentation is through directly using properties for characterizing subgraphs, such that the characterized subgraphs can be kept implicit as much as possible. Now, let us introduce this approach under different semantics.

First, under admissible semantics, according to **Prop1** and **Prop2**, we have the following proposition.

Proposition 6. *Let $G^p = (A, R, p)$ be a PrAG, and $E \subseteq A$ be a conflict-free set of arguments. It holds that:*

$$p(E^{ad}) \quad = \quad \Pi_{\alpha \in E} p(\alpha) \times \Pi_{\beta \in E_G^- \setminus E_G^+} p(\bar{\beta})$$

Proof. According to formulas (8) and (13), $p(E^{ad}) = \Sigma_{G' \in \rho^{ad}(E)} \, p(G') = \Sigma_{B \in 2^{I \cup E_G^+}} \, p(G_{\downarrow E \cup B})$. Since in $G_{\downarrow E \cup B}$,

- every argument in E appears,
- every argument in $E_G^- \backslash E_G^+$ does not appear,
- every argument in B appears, and
- and every argument in $(I \cup E_G^+) \backslash B$ does not appear,

it holds that $p(G_{\downarrow E \cup B}) = \Pi_{\alpha \in E} p(\alpha) \times \Pi_{\beta \in E_G^- \backslash E_G^+} p(\bar{\beta}) \times \Pi_{\gamma \in B} p(\gamma) \times \Pi_{\eta \in (I \cup E_G^+) \backslash B} p(\bar{\eta})$. Since $\Sigma_{B \in 2^{I \cup E_G^+}} (\Pi_{\gamma \in B} p(\gamma) \times \Pi_{\eta \in (I \cup E_G^+) \backslash B} p(\bar{\eta})) = 1$, we may conclude that:

$$\begin{aligned}
p(E^{ad}) &= \Sigma_{B \in 2^{I \cup E_G^+}} p(G_{\downarrow E \cup B}) \\
&= \Sigma_{B \in 2^{I \cup E_G^+}} (\Pi_{\alpha \in E} p(\alpha) \times \Pi_{\beta \in E_G^- \backslash E_G^+} p(\bar{\beta}) \times \Pi_{\gamma \in B} p(\gamma) \times \Pi_{\eta \in (I \cup E_G^+) \backslash B} p(\bar{\eta})) \\
&= (\Pi_{\alpha \in E} p(\alpha) \times \Pi_{\beta \in E_G^- \backslash E_G^+} p(\bar{\beta})) \times \Sigma_{B \in 2^{I \cup E_G^+}} (\Pi_{\gamma \in B} p(\gamma) \times \Pi_{\eta \in (I \cup E_G^+) \backslash B} p(\bar{\eta})) \\
&= \Pi_{\alpha \in E} p(\alpha) \times \Pi_{\beta \in E_G^- \backslash E_G^+} p(\bar{\beta}) \times 1 \\
&= \Pi_{\alpha \in E} p(\alpha) \times \Pi_{\beta \in E_G^- \backslash E_G^+} p(\bar{\beta})
\end{aligned}$$

Second, under stable semantics, according to **Prop1** and **Prop3**, we have the following proposition.

Proposition 7. *Let $G^p = (A, R, p)$ be a PrAG, and $E \subseteq A$ be a conflict-free set of arguments. It holds that:*

$$p(E^{st}) = \Pi_{\alpha \in E} p(\alpha) \times \Pi_{\beta \in I \cup (E_G^- \backslash E_G^+)} p(\bar{\beta})$$

The proof of Proposition 7 is similar to that of Proposition 6, so it is omitted.

Third, under complete semantics, according to **Prop1**, **Prop2** and **Prop4**, we have the following proposition.

Proposition 8. *Let $G^p = (A, R, p)$ be a PrAG, and $E \subseteq A$ be a conflict-free set of arguments. It holds that:*

$$\begin{aligned}
p(E^{co}) &= P_E \times P_{I_CO}, \textit{where} \\
P_E &= \Pi_{\alpha \in E} p(\alpha) \times \Pi_{\beta \in E_G^- \backslash E_G^+} p(\bar{\beta}), \textit{and} \\
P_{I_CO} &= \Sigma_{B' \in 2^I \wedge (\forall \alpha \in B' : \alpha_G^- \cap B' \neq \emptyset)} (\Pi_{\gamma \in B'} p(\gamma) \times \Pi_{\xi \in I \backslash B'} p(\bar{\xi}))
\end{aligned}$$

Proof. According to formulas (10) and (13), $p(E^{co}) = \Sigma_{G' \in \rho^{co}(E)} p(G') = \Sigma_{(B \in 2^{I \cup E_G^+}) \wedge (\forall \alpha \in B' : \alpha_G^- \cap B' \neq \emptyset)} p(G_{\downarrow E \cup B})$. Let $B' = B \cap I$ and $B'' = B \cap E_G^+$. It holds that $B = B' \cup B''$ and $B' \cap B'' = \emptyset$. Since in $G_{\downarrow E \cup B' \cup B''}$,

- every argument in E appears,
- every argument in $E_G^- \backslash E_G^+$ does not appear,
- every argument in B' (resp. B'') appears,
- and every argument in $I \backslash B'$ (resp. $E_G^+ \backslash B''$) does not appear,

it holds that $G_{\downarrow E \cup B} = G_{\downarrow E \cup B' \cup B''} = (\Pi_{\alpha \in E} p(\alpha) \times \Pi_{\beta \in E_G^- \backslash E_G^+} p(\bar{\beta}) \times \Pi_{\gamma \in B'} p(\gamma) \times \Pi_{\xi \in I \backslash B'} p(\bar{\xi}) \times \Pi_{\zeta \in B''} p(\zeta) \times \Pi_{\eta \in E_G^+ \backslash B''} p(\bar{\xi}))$. Since $\Sigma_{B'' \in 2^{E_G^+}} (\Pi_{\zeta \in B''} p(\zeta) \times \Pi_{\eta \in E_G^+ \backslash B''} p(\bar{\eta})) = 1$, we may conclude that:

$$
\begin{aligned}
p(E^{co}) &= \Sigma_{(B \in 2^{I \cup E_G^+}) \wedge (\forall \alpha \in B' : \alpha_G^- \cap B' \neq \emptyset)} p(G_{\downarrow E \cup B}) \\
&= \Sigma_{(B' \in 2^I) \wedge (B'' \in 2^{E_G^+}) \wedge (\forall \alpha \in B' : \alpha_G^- \cap B' \neq \emptyset)} p(G_{\downarrow E \cup B' \cup B''}) \\
&= \Sigma_{(B' \in 2^I) \wedge (B'' \in 2^{E_G^+}) \wedge (\forall \alpha \in B' : \alpha_G^- \cap B' \neq \emptyset)} (\Pi_{\alpha \in E} p(\alpha) \times \Pi_{\beta \in E_G^- \backslash E_G^+} p(\bar{\beta}) \times \\
&\quad \Pi_{\gamma \in B'} p(\gamma) \times \Pi_{\xi \in I \backslash B'} p(\bar{\xi}) \times \Pi_{\zeta \in B''} p(\zeta) \times \Pi_{\eta \in E_G^+ \backslash B''} p(\bar{\xi})) \\
&= (\Pi_{\alpha \in E} p(\alpha) \times \Pi_{\beta \in E_G^- \backslash E_G^+} p(\bar{\beta})) \times \Sigma_{B'' \in 2^{E_G^+}} (\Pi_{\zeta \in B''} p(\zeta) \times \Pi_{\eta \in E_G^+ \backslash B''} p(\bar{\eta})) \times \\
&\quad \Sigma_{(B' \in 2^I) \wedge (\forall \alpha \in B' : \alpha_G^- \cap B' \neq \emptyset)} (\Pi_{\gamma \in B'} p(\gamma) \times \Pi_{\xi \in I \backslash B'} p(\bar{\xi})) \\
&= P_E \times 1 \times P_{I_CO} \\
&= P_E \times P_{I_CO}
\end{aligned}
$$

Third, under preferred semantics, according to **Prop1**, **Prop2**, **Prop4** and **Prop5**, we have the following proposition.

Proposition 9. *Let $G^p = (A, R, p)$ be a PrAG, and $E \subseteq A$ be a conflict-free set of arguments. It holds that:*

$$
\begin{aligned}
p(E^{pr}) &= P_E \times P_{I_PR}, \textit{where} \\
P_E &= \Pi_{\alpha \in E} p(\alpha) \times \Pi_{\beta \in E_G^- \backslash E_G^+} p(\bar{\beta}), \textit{and} \\
P_{I_PR} &= \Sigma_{(B' \in 2^I) \wedge (\forall \alpha \in B' : \alpha_G^- \cap B' \neq \emptyset) \wedge (\mathcal{E}_{ad}(G_{\downarrow B'}) = \{\emptyset\})} (\Pi_{\gamma \in B'} p(\gamma) \times \Pi_{\xi \in I \backslash B'} p(\bar{\xi}))
\end{aligned}
$$

The proof of Proposition 9 is similar to that of Proposition 8, omitted.

Third, under grounded semantics, according to **Prop1**, **Prop2**, **Prop4** and **Prop6**, we have the following proposition.

Proposition 10. *Let $G^p = (A, R, p)$ be a PrAG, and $E \subseteq A$ be a conflict-free set of arguments. It holds that:*

$$\begin{aligned} p(E^{gr}) &= p(E^{co}) \times P_{GR}, \textit{where} \\ P_{GR} &= \Sigma_{(B''_2 \in 2^{E^+_G \cap E^-_G}) \wedge (\mathcal{E}_{gr}(G_{\downarrow E \cup B''}) = \{\{E\}\})} (\Pi_{\alpha \in B''} p(\alpha) \times \Pi_{\beta \in (E^+_G \cap E^-_G) \setminus B''} p(\bar{\beta})) \end{aligned}$$

The proof of Proposition 10 is similar to that of Proposition 8. The difference is that B'' is divided into two parts: B''_1 and B''_2, in which the arguments in B''_1 appear without constraints while the arguments in B''_2 appear only when $\mathcal{E}_{gr}(G_{\downarrow E \cup B''}) = \{\{E\}\}$ is satisfied. This is reflected by the factor P_{GR} and an equation $\Sigma_{B''_1 \in 2^{E^+_G \setminus E^-_G}} (\Pi_{\alpha \in B''_1} p(\alpha) \times \Pi_{\beta \in (E^+_G \setminus E^-_G) \setminus B''_1} p(\bar{\beta})) = 1$.

5. Algorithms and empirical results

The theoretical results presented in the previous section show that our characterized subgraphs based approach (called *C-Sub approach*) could be more efficient than the possible worlds based approach (called *PW approach*). In order to quantitatively evaluate the performance of our approach, in this section, by taking the cases under preferred semantics as an example, we first develop two algorithms for the PW approach and the C-Sub approach under preferred semantics respectively, and then conduct experiments to obtain the empirical results[5].

[5]The reasons why we choose preferred semantics for our empirical study here are as follows. First, efficiency of our approaches are mainly affected by the size of the set of remaining arguments, which is mainly dependent on the structure of graphs and the size of the extension, rather than on the semantics we choose. Second, preferred semantics has been widely used in many experiments (e.g., [15], [16] and [17], among others). And, we have also conducted several experiments under preferred semantics [18, 19]. Third, for some other properties, we will study them in our future work. For instance, under grounded semantics, an additonal property is how the efficiency of the new approach is affected by the size of $E^+_G \cap E^-_G$.

5.1. Algorithm for the PW approach under preferred semantics

Alg. 1 is an algorithm for the PW approach under preferred semantics. In this algorithm, up to 2^n subgraphs are blindly constructed where n is the number of nodes of the PrAG. For each subgraph $G_{\downarrow A'}$, if $E \subseteq A'$, then whether E is one of its preferred extensions is verified by the procedure *verify_preferred_labelling*$(\mathcal{L}, E)$. This procedure is based on the MC algorithm introduced in Section 2.

$\mathcal{L}$ is initialized as $(A', \emptyset, \emptyset)$, i.e. all arguments in A' are all labelled IN. Then, the procedure first checks whether there is an argument α in E such that α is super-illegally IN with respect to $\mathcal{L}$. If so, E is not a preferred extension. Otherwise, there are two possible cases. First, no argument is illegally IN. It follows that $in(\mathcal{L})$ is admissible. In this case, if $E \subset in(\mathcal{L})$, then E is not a preferred extension. Second, there are some arguments that are illegally IN. In this case, the procedure iteratively selects arguments that are illegally IN (or *super-illegally IN*) and applies a *transition step* to obtain a new labelling, until a labelling is reached in which no argument is illegally IN.

5.2. Algorithm for the C-Sub approach under preferred semantics

Alg. 2 is an algorithm for the C-Sub approach under preferred semantics. Unlike the PW approach, the algorithm first gets a set of remaining argument $I = A \backslash (E^- \cup E^+ \cup E)$. Then, for each subset B' of I, verify whether the subgraph induced by B' has an nonempty admissible extension. The procedure *verify_nonempty_adm*$((\mathcal{L})$ recursively selects arguments that are illegally IN (or *super-illegally IN*) and applies a *transition step* to obtain a new labelling, until a lablling is reached in which no argument is illegally IN. If there is a labeling $\mathcal{L}$ such that $\mathcal{L}$ has no argument that is illegally IN and $in(\mathcal{L}) \neq \emptyset$, then the procedure returns true. Otherwise, it returns false.

Then, $p(E^{pr})$ is computed according to Proposition 9. More specifically, in Steps 11 and 12, P_E is computed; from Step 13 to Step 17, P_{I_PR} and $p(E^{pr})$ are computed.

5.3. Empirical results

The algorithms were implemented in Java, and tested on a machine with an Intel CPU running at 2.26 GHz and 2.00 GB RAM. We conducted three experiments to test the performance of out C-Sub approach.

Alg. 1 Algorithm for the PW approach under preferred semantics

input : $G^p = (A, R, p)$ and a conflict-free set $E \subseteq A$
output: $p(E^{pr})$

$p(E^{pr}) := 0$;
for each $A' \in 2^A$ and $A' \neq \emptyset$ do
    if $E \subseteq A'$ then
        $\mathcal{L} := (A', \emptyset, \emptyset)$;
        if *verify_preferred_labelling*$(\mathcal{L}, E) = true$ then
            $p(E^{pr}) := p(E^{pr}) + p(G_{\downarrow A'})$;
        end if
    end if
end do

procedure *verify_preferred_labelling*$(\mathcal{L}, E)$
$vpr := true$;
if $\mathcal{L}$ has an argument α that is super-illegally IN and $\alpha \in E$ **then**
    $vpr := false$;
else
    if $\mathcal{L}$ does not have an argument that is illegally IN **then**
        if $E \subset in(\mathcal{L})$ **then** $vpr := false$; **end if**
    else
        if $\mathcal{L}$ has an argument that is super-illegally IN **then**
            $\alpha :=$ some argument that is super-illegally IN in $\mathcal{L}$;
            verify_preferred_labelling$(transition_step(\mathcal{L}, \alpha), E)$;
        else
            for each α that is illegally IN in $\mathcal{L}$ **do**
                verify_preferred_labelling$(transition_step(\mathcal{L}, \alpha), E)$
            end for
        end if
    end if
end if
return vpr
end procedure

Alg. 2 Algorithm for C-Sub approach under preferred semantics

```
input : G^p = (A, R, p), a conflict-free set E ⊆ A
output: p(E^pr)
p(E^pr) := 0; sub := ∅; I := A\(E^− ∪ E^+ ∪ E); x_1 := 1; x_2 := 1;
for each B' ∈ 2^I do
    if B' = ∅ then sub := sub ∪ {∅};
    else
        if there exists no α ∈ B' such that α^−_G ∩ B' = ∅ then
            L := (B', ∅, ∅);
            if verify_nonempty_adm(L) = false then sub := sub ∪ {B'}; end if
        end if
    end if
end for
for each α ∈ E^−_G\E^+_G do x_1 := x_1 × p(ᾱ); end for
for each α ∈ E do x_1 := x_1 × p(α); end for
for each B' ∈ sub do
    for each α ∈ I\B' do x_2 := x_2 × p(ᾱ); end for
    for each α ∈ B' do x_2 := x_2 × p(α); end for
    p(E^pr) := p(E^pr) + x_1 × x_2;
end for

procedure verify_nonempty_adm(L)
vna := false;
if L has no argument that is illegally IN then
    if in(L) ≠ ∅ then vna := true;
else
    if L has an argument that is super-illegally IN then
        α := some argument that is super-illegally IN in L;
        verify_nonempty_adm(transition_step(L, α));
    else
        for each α that is illegally IN in L do
            verify_nonempty_adm(transition_step(L, α));
        end for
    end if
end if
return vna;
end procedure
```

The first experiment is about the average computation time of the C-Sub approach and that of the PW approach, according to the following configuration of PrAGs:

- The number of nodes of PrAGs is from 10 to 25 (since when the number of nodes is smaller than 10, the computation time of the two approaches is close to 0 millisecond, while the number of nodes is bigger than 25, the computation time of the PW approach is almost always more than 3 minutes which we set as the point of timeout).

- The ratios of the number of edges to the number of nodes are 1:1, 2:1 and 3:1 respectively (since the density of PrAGs is an important factor affecting the average computation time of the two approaches).

- The size of the extension is 3. This number is selected somewhat arbitrarily. How the size of the extension affects the average computation time of the two approaches will be studied in another experiment.

This configuration consists of $16 \times 3 \times 1 = 48$ assignments for the two approaches respectively. Each assignment is a tuple $(\#\text{nodes}, i:1, j)$, where "#nodes" is the number of nodes, $i:1$ is the ratio of the number of edges to the number of nodes, and j is the size of the extension. For convenience, we use PW_$j[i:1]$ (C-Sub_$j[i:1]$) to denote the (average) computation time of the PW approach (resp. the C-Sub approach) when the size of the extension is j and the the ratio of the number of edges to the number of nodes is $i:1$, and the number of nodes is given.

For each assignment, the algorithms were executed 20 times respectively. In each time, a PrAG (including its notes, edges, and the probabilities of nodes) and a conflict-free set E of arguments were generated at random. For simplicity, the probabilities assigned to nodes are nonzero. Then, the probability of E being a preferred extension was computed by the PW approach and the C-Sub approach respectively. Table 3 shows the average execution time of the two approaches.

Since in many cases, the execution time might last very long, to make the test possible, when the time for computing $p(E^{pr})$ is over 3 minutes (180 seconds), the execution was stopped by setting a break in the program. When the number of timeout is less than 20, the average time was recorded, and for each timeout, the time used for calculation is 180 seconds. For instance, when #nodes = 25, C-Sub_3 [3] = 15.828 seconds. The detailed records of 20

#nodes	PW_3 [1:1] (secs/ timeout)	PW_3 [2:1] (secs/ timeout)	PW_3 [3:1] (secs/ timeout)	C-Sub_3 [1:1] (secs/ timeout)	C-Sub_3 [2:1] (secs/ timeout)	C-Sub_3 [3:1] (secs/ timeout)
10	0.015/0	0.120/0	0.585/0	0.001/0	0.000/0	0.000/0
11	0.039/0	0.648/0	4.346/0	0.000/0	0.000/0	0.000/0
12	0.070/0	3.141/0	34.662/0	0.002/0	0.001/0	0.000/0
13	0.160/0	6.732/0	101.667/6	0.006/0	0.005/0	0.000/0
14	0.380/0	23.879/1	154.271/13	0.000/0	0.000/0	0.001/0
15	4.772/0	87.185/8	/20	0.003/0	0.002/0	0.002/0
16	2.236/0	112.569/11	/20	0.001/0	0.003/0	0.000/0
17	11.674/0	107.201/9	/20	0.003/0	0.003/0	0.001/0
18	18.445/1	149.583/16	/20	0.012/0	0.003/0	0.019/0
19	31.282/1	159.580/17	/20	0.015/0	0.011/0	0.023/0
20	50.973/2	/20	/20	0.028/0	0.477/0	0.211/0
21	89.654/5	/20	/20	0.088/0	0.067/0	11.665/1
22	143.039/10	/20	/20	0.106/0	14.925/1	10.043/1
23	/20	/20	/20	0.627/0	6.901/0	13.825/1
24	/20	/20	/20	3.067/0	12.434/1	0.741/0
25	/20	/20	/20	1.406/0	30.222/3	15.429/1

Table 3: The average execution time of the PW approach and the C-Sub approach

No.	C-Sub_3 [3:1] (#nodes = 25)			C-Sub_3 [2:1] (#nodes = 25)		
	time (secs)	max $\|B'\|$	avg. $\|B'\|$	time (secs)	max $\|B'\|$	avg. $\|B'\|$
1	0.016	8	4	**timeout**	**15**	**1**
2	3.760	12	6	0.640	15	7
3	0.000	10	5	0.000	10	5
4	0.015	7	3	0.281	16	8
5	**22.074**	**11**	**5**	0.000	11	5
6	0.016	10	5	4.339	13	6
7	**24.039**	**11**	**5**	**57.424**	**14**	**7**
8	0.078	11	5	0.078	14	7
9	**timeout**	**13**	**5**	**timeout**	**16**	**1**
10	4.524	11	5	0.000	10	5
11	**43.275**	**14**	**7**	0.031	13	6
12	0.000	9	4	0.047	14	7
13	0.000	9	4	0.125	15	7
14	0.016	9	4	0.047	14	7
15	0.000	11	5	1.310	13	6
16	0.015	11	5	**timeout**	**16**	**3**
17	0.000	9	4	0.047	13	6
18	**30.732**	**12**	**6**	0.031	12	6
19	0.016	9	4	0.016	12	6
20	0.000	10	5	0.062	11	5
avg.	15.429	10.35		30.222	13.35	

Table 4: The detailed records of the execution time of the C-Sub approach. In this table, "max $|B'|$" and "avg. $|B'|$" denote respectiely the maximal and average size of B'.

times of execution are shown in Table 4. For instance, C-Sub_3 [3] = (0.016 + 3.760 + 0.000 + 0.015 + 22.074 + 0.016 + 24.039 + 0.078 + 180 + 4.524 + 43.275 + 0.000 + 0.000 + 0.016 + 0.000 + 0.015 + 0.000 + 30.732 + 0.016 + 0.000) ÷20 = 15.429.

From Table 3, we found that the C-Sub approach greatly outperforms the PW approach. The computation time of the PW approach increases dramatically with the increase of the number of nodes and the density of edges. More specifically, when the number of nodes is given, PW_3 [i : 1] increases sharply with the increase of i. For instance, when #nodes = 15, PW_3 [1 : 1] = 4.772, PW_3 [2 : 1] = 87.185 (with 8 timeouts), and PW_3

[3 : 1] has no record of time (with 20 timeouts). Meanwhile, when the density of edges is given, PW_3 [i : 1] ($i = 1, 2, 3$) increases exponentially with the increase of #nodes. On the contrary, with the increase of density (i.e., i : 1), C-Sub_3 [i : 1] might not increase. And, with the increase of the number of nodes, PW_3 [i : 1] ($i = 1, 2, 3$) does not increase exponentially. The basic reason behind these phenomena is that according to the theoretical results obtained in Section 4, compared to the PW approach, the complexity of the C-Sub approach decreases from $|2^A|$ to $|2^I|$. In other words, the complexity of the C-Sub approach is manly determined by the size of I (i.e., the maximal size of B'). This is evidenced by the data shown in Table 4, in which the average value of maximal sizes of B' in 20 tests is 10.35 for C-Sub_3 [3 : 1] and 13.35 for C-Sub_3 [2 : 1], which matches very well to the average computation time of C-Sub_3 [3 : 1] (15.429 seconds) and C-Sub_3 [2 : 1] (30.222 seconds).

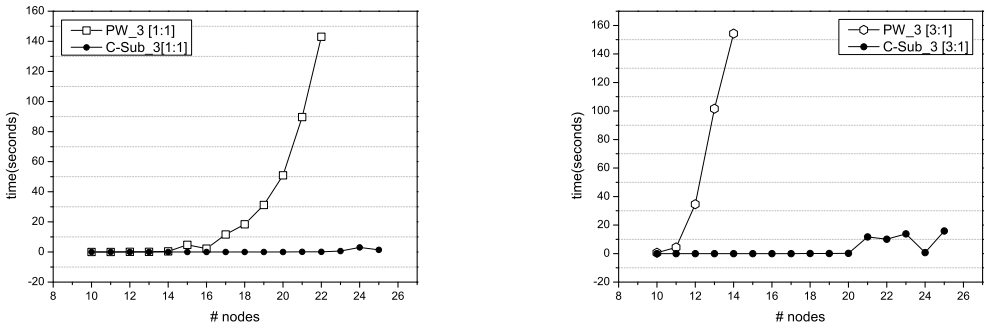


Figure 6: Plots showing the execution time of the PW approach and the C-Sub approach

The second experiment is to further study how the increase of density of PrAGs affects the computation time of the two approaches.

As shown in Table 5 and Figure 7, the configuration for the PW approach is: the size of extension is 3, the number of nodes is 10, and the density of edges ranges from 1 : 1 to 6 : 1. The configuration of the C-Sub approach is similar to that of the PW approach, the only difference is that the number of nodes is 20 for the C-Sub approach (in that when the number of nodes is less than 10, the average computation time of the PW approach is close to 0). The average execution time of the PW approach increases sharply with

$i:1$	1:1	2:1	3:1	4:1	5:1	6:1
PW_3 $[i:1]$ (secs) #nodes = 10	0.026	0.141	0.694	1.641	2.781	4.056
C-Sub_3$[i:1]$ (secs) #nodes = 20	0.255	0.367	0.090	0.014	0.003	0.001

Table 5: Average execution time of the PW approach and the C-Sub approach w.r.t. the changing of density of edges

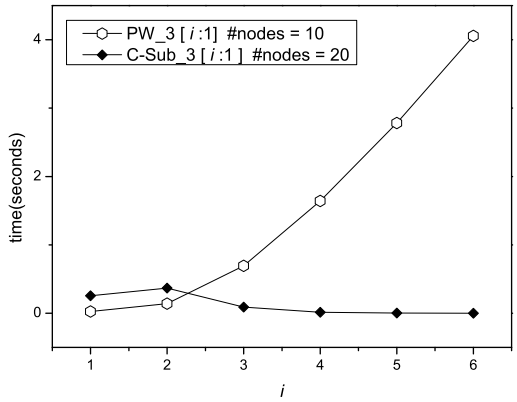


Figure 7: Plot showing the average execution time of the PW approach and the C-Sub approach w.r.t. the changing of density of edges

the increase of the density of edges, while the execution time of the C-Sub approach decreases with the increase of the density of edges.

The third experiment is study how the average execution time of the PW approach and the C-Sub approach changes with respect to the changing of the size of the extension. In this experiment, the configuration for the two approaches is: the size of extension is 3 and 5, the number of nodes ranges from 10 to 25, and the density of edges is $2:1$.

According to the results shown Figure 8 (corresponding to the data in Table 6), the shapes of the graphs PW_3 [2:1] and PW_5 [2:1] are almost the same, which means that the average execution time of the PW approach does not fundamentally decrease with the changing of the size of the extension. On the contrary, the average execution time of the C-Sub approach decreases

#nodes	PW_3 [2:1] (secs/timeout)	PW_5 [2:1] (secs/timeout)	C-Sub_3 [2:1] (secs/timeout)	C-Sub_5 [2:1] (secs/timeout)
10	0.120/0	0.018/0	0.000/0	0.000/0
11	0.648/0	0.042/0	0.000/0	0.000/0
12	3.141/0	0.125/0	0.001/0	0.000/0
13	6.732/0	0.368/0	0.005/0	0.000/0
14	23.879/1	4.178/0	0.000/0	0.000/0
15	87.185/8	3.723/0	0.002/0	0.000/0
16	112.569/11	43.981/2	0.003/0	0.002/0
17	107.201/9	70.016/5	0.003/0	0.000/0
18	149.583/16	93.756/8	0.003/0	0.000/0
19	159.580/17	108.857 /10	0.011/0	0.003/0
20	/20	151.422/15	0.477/0	0.000/0
21	/20	155.704/16	0.067/0	0.000/0
22	/20	171.270/16	14.925/1	0.003/0
23	/20	/20	6.091/0	0.001/0
24	/20	/20	12.434/1	0.222/0
25	/20	/20	30.222/3	0.008/0

Table 6: The average execution time of the PW approach and the C-Sub approach with respect to different sizes of the extension

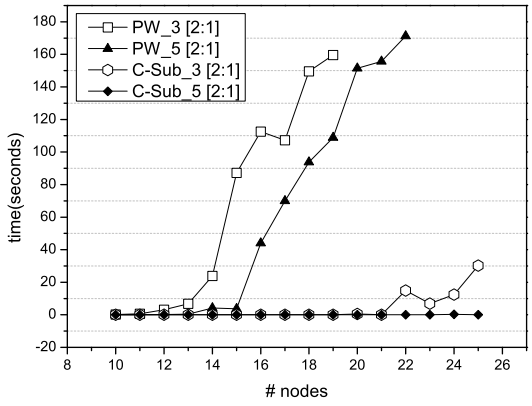


Figure 8: Plots showing the average execution time of the PW approach and the C-Sub approach with respect to different sizes of the extension

to a great extent. The basic reason behind this phenomenon is that: since the complexity of the C-Sub approach is manly determined by the size of $I = A \backslash (E \cup E_G^+ \cup E_G^-)$, with the increase of the size the extension E, the size of I become smaller.

6. Computational properties

Based on the theory and the experimental results introduced in Sections 4 and 5, in this section, we briefly analyze some computational properties of our C-Sub approach (or briefly "our approach").

On the one hand, according to *classical complexity theory*, by using the C-Sub approach, it holds that computing $p(E^{ad})$ and $p(E^{st})$ is polynomial time tractable, while under complete, preferred and grounded semantics, problems of determining $p(E^{co})$, $p(E^{pr})$ and $p(E^{gr})$ are still intractable. This is because: under complete and preferred semantics, we need to consider $|2^I|$ cases, while under grounded semantics, $|2^I| + |2^{E_G^+ \cap E_G^-}|$ cases. However, theoretically, the C-Sub approach is more efficient, in that:

1. most subgraphs are not necessary to be constructed and computed, i.e., the maximal number of subgraphs decreases from $|2^A|$ to $|2^I|$ (or $|2^I| + |2^{E_G^+ \cap E_G^-}|$), where $I = A \backslash (E \cup E_G^+ \cup E_G^-)$ is the set of remaining arguments; and
2. the size of the maximal subgraph decreases from $|A|$ to $|I|$ (or $|E_G^+ \cap E_G^-|$).

The efficiency of the C-Sub approach is evidenced by the empirical results. This approach not only dramatically decreases the time for computing $p(E^\sigma)$, but also has an attractive property, which is contrary to that of existing approaches: the denser the edges of a PrAG are or the bigger the size of a given extension E is, the more efficient our approach computes $p(E^\sigma)$.

On the other hand, under complete and preferred semantics, since the complexity of the C-Sub approach is mainly determined by the size of the remaining arguments, which is usually much smaller than that of the whole set of arguments in a PrAG, according to *parameterized complexity theory*, the problems of determining $p(E^{co})$ and $p(E^{pr})$ in the C-Sub approach are fixed-parameter tractable with respect to the size of remaining arguments. Details are as follows.

In terms of parameterized complexity theory, *the complexity of a problem is not only measured in terms of the input size, but also in terms of a parameter. The theory's focus is on situations where the parameter can be assumed to be small* [20]. Let Q be a classical problem, and k be a parameter of the problem. A parameterized problem is denoted as (Q, k). When binding k to a fixed constant, in many cases, an intractable problem Q can be made tractable. This property is called *fixed-parameter tractability* (FPT). More specifically, let n be the input size of a problem, and f be a computable function that depends on a parameter k of the problem. The complexity class FTP consists of problems that can be computed in $f(k) \cdot n^{\mathcal{O}(1)}$.

In the setting of this paper, let $k = |I|$. Typically, k is much smaller than the size of the PrAG (i.e., $|A|$). Under complete and preferred semantics, the complexity of determining the probability that a set of arguments is an extension is dominated by the size of the set of remaining arguments (i.e., $|I|$). Formally, we have the following proposition.

Proposition 11. *Let $G^p = (A, R, p)$ be a PrAG, and $E \subseteq A$ be a conflict-free set of arguments. Let $k = |I|$ where I is the set of remaining arguments of G with respect to E. Let $Pr_G(E^{co})$ and $Pr_G(E^{pr})$ be the problems of determining the probability $p(E^{co})$ and $p(E^{pr})$ respectively. It holds that $(Pr_G(E^{co}), k)$ and $(Pr_G(E^{pr}), k)$ belong to FTP.*

Proof. First, under preferred semantics, the algorithm (Alg. 2) consists of the following two parts. The first part (Lines 2 - 10; Lines 18 - 33) is the difficult core of the algorithm. In this part, there are 2^k calls and in each call, the procedure *verify_nonempty_adm*$(\mathcal{L})$ may be intractable. However, since the size of the subgraphs induced by B' is less than k, the time for executing *verify_nonempty_adm*$(\mathcal{L})$ is dependent on k, denoted as $g(k)$. The second part (Line 1, Lines 11 - 17) is tractable. The execution time of this part can be bounded by $n^{\mathcal{O}(1)}$ where $n = |A|$. So, the overall execution time can be bounded by $2^k \cdot g(k) + n^{\mathcal{O}(1)} = f(k) + n^{\mathcal{O}(1)} \leqslant f(k) \cdot n^{\mathcal{O}(1)}$ where $f(k) = 2^k \cdot g(k)$. Hence, $(Pr_G(E^{pr}), k)$ belongs to FTP.

Second, in terms of Propositions 8 and 9, the algorithm under complete semantic (not presented in the present paper) is similar to the one under preferred semantics. The difference is that under complete semantics, *verify_nonempty_adm*$(\mathcal{L})$ is not executed. So, it holds that $(Pr_G(E^{co}), k)$ belongs to FTP.

Note that under grounded semantics, since usually it may be not the case that $\max\{|E_G^+ \cap E_G^-|, |I|\}$ is much smaller than $|A|$, Proposition 11 can not be applied to grounded semantics.

7. Related work

In this paper, we have proposed a new approach (the C-Sub approach) to formulate semantics of probabilistic argumentation, and analyzed its computational properties on the basis of an empirical study. To the best of our knowledge, our approach is the first attempt to systematically study how to compute the semantics of probabilistic argumentation without (or with less) construction and computation of subgraphs not only under admissible and stable semantics, but also under other semantics including complete, grounded and preferred. In this section, we give a discussion about some related work.

Independence assumption of arguments. In this paper, we assume the independence of arguments appearing in a graph. A theoretical foundation for this assumption is originally formulated by Anthony Hunter in a series of his work [7, 21, 6] from the justification perspective on the probability of an argument:

> *For an argument α in a graph G, with a probability assignment p, $p(\alpha)$ is treated as the probability that α is a justified point (i.e. each is a self-contained, internally valid, contribution) and therefore should appear in the graph, and $1-p(\alpha)$ is the probability that α is not a justified point and so should not appear in the graph. This means the probabilities of the arguments being justified are independent (i.e., knowing that one argument is a justified point does not affect the probability that another is a justified point).*

The justification perspective can be further illustrated by the following example that was originally presented in [7]. Given two arguments $\alpha_1 = (\{p\}, p)$ and $\alpha_2 = (\{\neg p\}, \neg p)$ constructing from a knowledge base containing just two formulae $\{p, \neg p\}$, α_1 attacks α_2 and vice versa. In terms of classical logic, it is not possible that both arguments are true, but each of them is a justified point. So even though logically α_1 and α_2 are not independent (in the sense that if one is known to be true, then the other is known to be false), they are independent as justified points.

In existing literature, some models and algorithms depend on an independence assumption of arguments and/or attacks [4, 9, 7, 21, 8], while others do not [22, 3, 6, 23]. For the former, probabilities are assigned to arguments and/or attacks, and the probability distribution over subgraphs can be generated based on the independence assumption. For the latter, users directly specify the unique probability distribution over the set of subgraphs. There are pros and cons about whether the independence assumption is used or not. On the one hand, by using the independence assumption, it can be more efficient to use the probability assignment to arguments and/or attacks and then generate the probability distribution over subgraphs. But, as discussed in [6], whilst the independence assumption is useful in some situations, it is not always appropriate. On the other hand, when the independence assumption is avoided, the dependence relation between arguments can be properly represented. However, in this way, users have to specify probability distribution over the set of subgraphs, whose number is exponential with respect to the arguments and/or attacks. And, in many cases users may not be aware of the probability value that should be assigned to a possible world (subgraph) which may represent a complex scenario [24]. In this sense, in many situations, the models without independence assumption might not be applicable.

So, with regard to whether an independence assumption is used or not, there are both advantages and disadvantages.

Complexity analysis and algorithms for probabilistics argumentation. Computational issues of probabilistic argumentation have been deeply investigated in recent years.

On the one hand, Fazzinga et al studied the complexity problem of determining the probability that a set of arguments is an extension under a given semantics [9]. The results show that under admissible and stable semantics, the problem belongs to $PTIME$, while under complete, grounded, preferred and ideal semantics, the problem is $FP^{\sharp P}$. However, the existing work only studied the complexity problems from the perspective of classical complexity theory. The corresponding problems from the perspective of parameterized complexity theory have not been explored.

On the other hand, since using a brute-force algorithm to evaluate the probability of a set of arguments being an extension is computationally prohibitive, in existing work, an approximate approach (called the Monte-Carlo simulation approach) has been proposed to cope with this problem [4], which

was significantly improved in [8] by reducing the sample space of computation. Corresponding to these approximate approaches, however, little attention has been paid to the development of exact approaches. Our approach presented in this paper is a first step in this direction.

Efficient algorithms based on the structural properties of graphs. Since an abstract argumentation framework (argument graph) can be viewed as a digraph, applying various properties of existing graph theory to argumentation is not new. For instance, when an argument graph satisfies some properties (acyclic, symmetric, bipartite, etc), there exist tractable algorithms to compute its semantics [25, 26]; when an argument graph has bounded tree-width, there exist fixed-parameter algorithms [15]; when decomposing an argument graph based on the notion of strongly connected components, the efficiency of computation can be significantly improved [19]; by mapping the notion of a kernel, a semikernel and a maximal semikernel in a directed graph [27] respectively to the notion of a stable set, an admissible set and a preferred extension in an argumentation framework [34, 35], in terms of [28], we may infer that the complexity results and algorithms related to kernels and semikernels can be applied to formal argumentation. Beside the structural properties that have been applied to argumentation, notions of kernels and semikernels have also been connected to logic programs and default theories. According to [28], every normal logic program can be transformed into a graph. The stable, partial stable and well-founded semantics correspond to kernels, semikernels and the initial acyclic part, respectively. Meanwhile, according to [29], it is an equivalence relation between the problem of the existence of kernels in digraphs and satisfiability of propositional theories (SAT). Thanks to this relation, algorithms for computing kernels can be applied to computing the semantics of logic programs.

It is worth to note that although structural properties of digraphs have been exploited in Dung's abstract argumentation [1], we have not found solutions to apply these properties to the efficient computation of the semantics of probabilistic argumentation. So, in this paper, based on some basic structural properties of digraphs and formal argumentation [27, 1], we have defined properties that can be used to characterize subgraphs of a PrAG. These properties are established in the setting of probabilistic argumentation where the appearance of arguments is related to a given extension E, and on the basis of the original definition of extensions under different argumentation semantics [1]. Despite of their simplicity, these properties lay a

concrete foundation to define a new methodology to formulate and compute the semantics of probabilistic argumentation.

Kernelization and parameterized algorithms. The approach and results presented in this paper have a close relation to some existing work on kernelization and parameterized algorithms, which have been extensively studied in the past two decades. Kernelization is a systematic approach to study polynomial-time preprocessing algorithms, such that the "easy parts" of a problem instance can be solved efficiently, and the problem instance is reduced to its computationally difficult "core" structure (the problem kernel of the instance) [30]. If the size of the kernel can be effectively bounded in terms of a fixed-parameter alone, then the problem is *fixed-parameter tractable* (FPT) [20]. In recent years, fixed-parameter algorithms were developed in the setting of Dung's abstract argumentation, by exploiting some important parameters for graph problems, such as the *tree-width* [25, 15] and the *clique-width* [31, 32] of a graph.

The C-Sub approach presented in this paper can be understood as a kind of kernelization. The novelty of this approach lies in the fact that new properties are defined to characterize the subgraphs of a PrAG with respect to a given extension.

8. Conclusions and future work

Probabilistic argumentation is an emerging direction in the area of formal argumentation. In this paper, we have studied the formulation and computation of semantics of probabilistic argumentation. The main contributions of this paper are two-fold.

On the one hand, conceptually, we define specific properties to characterize the subgraphs of a PrAG with respect to a given extension, such that the probability of a set of arguments E being an extension can be defined in terms of these properties, without (or with less) construction of subgraphs. The theoretical results in this paper show that under admissible and stable semantics, computing a set of arguments being an extension of a PrAG is polynomial time tractable; under complete and preferred semantics, the problems of determining $p(E^{co})$ and $p(E^{pr})$ in our C-Sub approach are fixed-parameter tractable with respect to the size of remaining arguments.

On the other hand, computationally, we take preferred semantics as an example, and develop algorithms to evaluate the efficiency of our approach.

The empirical results show that our approach not only dramatically decreases the time for computing the semantics of probabilistic argumentation, but also has an attractive property, which is contrary to that of existing approaches: the denser the edges of a PrAG are or the bigger the size of a set of arguments E is, the more efficient our approach computes the probability of E being an extension of the PrAG.

Future work is as follows. First, in this paper we deal with the probabilistic argument graphs (PrAGs) in which probabilities are assigned to arguments. However, when probabilities are assigned to attacks or to both arguments and attacks, the formalisms and algorithms corresponding to the ones in this paper are expected to be different. In [6], only attacks are assigned with probabilities. So, it would be interesting to combine the theory presented in [6] with the approach presented in this paper. Meanwhile, one may consider to extend our approach to the cases where both arguments and attacks are assigned with probabilities, similar to the work presented in [8]. Second, an independence assumption of arguments is used in the paper. Although it is useful in some situations, but not always appropriate. So, a further step is to develop the corresponding approaches without this assumption. Third, as mentioned above, under grounded semantics, we have not obtained a conclusion that our approach is fixed-parameter tractable. Further analysis about this issue is needed. Fourth, in the empirical study, we have only considered preferred semantics. The algorithms and experiments under other semantics (especially grounded semantics) are also important.

Acknowledgment

We are grateful to the reviewers of this paper for their constructive and insightful comments. The research reported in this paper was partially supported by the National Research Fund Luxembourg (FNR) and Zhejiang Provincial Natural Science Foundation of China (No. LY14F030014).

References

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

[2] Dung, P.M., Thang, P.M.: Towards (probabilistic) argumentation for jury-based dispute resolution. In: Proceedings of the COMMA 2010, IOS Press (2010) 171–182

[3] Rienstra, T.: Towards a probabilistic dung-style argumentation system. In: Proceedings of the AT. (2012) 138–152

[4] Li, H., Oren, N., Norman, T.J.: Probabilistic argumentation frameworks. In: Proceedings of the TAFA 2011, Springer (2012) 1–16

[5] Dunne, P.E., Hunter, A., McBurney, P., Parsons, S., Wooldridge, M.: Weighted argument systems: Basic definitions, algorithms, and complexity results. Artificial Intelligence **175**(2) (2011) 457–486

[6] Hunter, A.: Probabilistic qualification of attack in abstract argumentation. International Journal of Approximate Reasoning **55**(2) (2014) 607–638

[7] Hunter, A.: Some foundations for probabilistic abstract argumentation. In: Proceedings of the 4th International Conference on Computational Models of Argument, IOS Press (2012) 117–128

[8] Fazzinga, B., Flesca, S., Parisi, F.: On efficiently estimating the probability of extensions in abstract argumentation frameworks. InternationalJournalofApproximateReasoning **69** (2016) 106–132

[9] Fazzinga, B., Flesca, S., Parisi, F.: On the complexity of probabilistic abstract argumentation. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, AAAI Press (2013) 898–904

[10] Liao, B., Huang, H.: Formulating semantics of probabilistic argumentation by characterizing subgraphs. In: Proceedings of the LORI 2015, Springer (2015) 243–254

[11] Baroni, P., Caminada, M., Giacomin, M.: An introduction to argumentation semantics. The Knowledge Engineering Review **26**(4) (2011) 365–410

[12] Jakobovits, H., Vermeir, D.: Robust semantics for argumentation frameworks. Journal of logic and computation **9**(2) (1999) 215–261

[13] Modgil, S., Caminada, M.: Proof theories and algorithms for abstract argumentation frameworks. In: I. Rahwan, G. R. Simari (eds.), Argumentation in Artificial Intelligence, Springer (2009) 105–129

[14] Baroni, P., Giacomin, M., Guida, G.: Scc-recursiveness: a general schema for argumentation semantics. Artificial Intelligence **168** (2005) 162–210

[15] Dvořák, W., Pichler, R., Woltran, S.: Towards fixed-parameter tractable algorithms for abstract argumentation. Artificial Intelligence **186** (2012) 1–37

[16] Cerutti, F., Giacomin, M., Vallati, M., Zanella, M.: An scc recursive meta-algorithm for computing preferred labellings in abstract argumentation. In: C. Baral, G.D. Giacomo, T. Eiter (Eds.), Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning, KR 2014, AAAI Press (2014) 42–51

[17] Nofal, S., Atkinson, K., Dunne, P.: Algorithms for decision problems in argument systems under preferred semantics. Artificial Intelligence **207** (2014) 723–51

[18] Liao, B., Huang, H.: Partial semantics of argumentation: Basic properties and empirical results. Journal of Logic and Computation **23**(3) (2013) 541–562

[19] Liao, B.: Toward incremental computation of argumentation semantics: A decomposition-based approach. Annals of Mathematics and Artificial Intelligence **67**(3-4) (2013) 319–358

[20] Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer (2006)

[21] Hunter, A.: A probabilistic approach to modelling uncertain logical arguments. International Journal of Approximate Reasoning **54** (2013) 47–81

[22] Thimm, M.: A probabilistic semantics for abstract argumentation. In: European Conference on Artificial Intelligence (ECAI), AAAI Press (2012) 750–755

[23] Grossi, D., van der Hoek, W.: Audience-based uncertainty in abstract argument games. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. (2013) 143–149

[24] Fazzinga, B., Flesca, S., Parisi, F.: On the complexity of probabilistic abstract argumentation frameworks. Transactions on Computational Logic **16**(3) (July 2015)

[25] Dunne, P.E.: Computational properties of argument systems satisfying graph-theoretic constraints. Artificial Intelligence **171**(10-15) (2007) 701–729

[26] Coste-Marquis, S., Devred, C., Marquis, P.: Symmetric argumentation frameworks. In: L. Godo (Ed.), Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005), Springer (2005) 317–328

[27] Galeana-Sánchez, H., Neumann-Lara, V.: On kernels and semikernels of digraph. Discrete Mathematics **48** (1984) 67–76

[28] Dimopoulos, Y., Torres, A.: Graph theoretical structures in logic programs and default theories. Theoretical Computer Science **170** (1996) 209–214

[29] Walicki, M., Dyrkolbotn, S.: Finding kernels or solving sat. Journal of Discrete Algorithms **10** (2012) 146–164

[30] Lokshtanov, D., Marx, D., Fomin, F.V., Kowalik, L., Pilipczuk, M., Cygan, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer (2015)

[31] Dvořák, S., Szeider, S., Woltran, S.: Reasoning in argumentation frameworks of bounded clique-width. In: P. Baroni, F. Cerutti, M. Giacomin, G.R. Simari (Eds.), Proceedings of the 3rd Conference on Computational Models of Argument, COMMA 2010, IOS Press (2010) 219–230

[32] Dvořák, W., Ordyniak, S., Szeider, S.: Augmenting tractable fragments of abstract argumentation. Artificial Intelligence **186** (2012) 157–173

[33] Dunne, P.E., Wooldridge, M.: Complexity of Abstract Argumentation, In: Argumentation in Artificial Intelligence, Springer US (2009) 85–104

[34] Coste-Marquis, S., Devred, C., Marquis, P.: Symmetric Argumentation Frameworks, In: Proceedings of the ECSQARU 2005, Springer (2005) 317–328

[35] Dyrkolbotn, S.: On a Formal Connection between Truth, Argumentation and Belief, In: Proceedings of the ESSLLI 2012/2013, LNCS 8607, Springer (2014) 69–90