# Probabilistic Abstract Argumentation Frameworks, A Possible World View

Theofrastos Mantadelis[1]

*Konstantinou Palaiologou 1, 1011 Nicosia, Cyprus*

Stefano Bistarelli[2]

*Via Vanvitelli 1, 06123 Perugia, Italy*

**Abstract**

After Dung's founding work in Abstract Argumentation Frameworks there has been a growing interest in extending the Dung's semantics in order to describe more complex or real life situations. Several of these approaches take the direction of weighted or probabilistic extensions. One of the most prominent probabilistic approaches is that of constellation Probabilistic Abstract Argumentation Frameworks.

In this paper, we first make the connection of possible worlds and constellation semantics; we then introduce the probabilistic attack normal form for the constellation semantics; we furthermore prove that the probabilistic attack normal form is sufficient to represent any Probabilistic Abstract Argumentation Framework of the constellation semantics; then we illustrate its connection with Probabilistic Logic Programming and briefly present an existing implementation. The paper continues by also discussing the probabilistic argument normal form for the constellation semantics and proves its equivalent properties. Finally, this paper introduces a new probabilistic structure for the constellation semantics, namely probabilistic cliques.

*Email addresses:* theofrastos.mantadelis@ouc.ac.cy (Theofrastos Mantadelis), stefano.bistarelli@unipg.it (Stefano Bistarelli)

[1]Open University of Cyprus & Research Center on Interactive Media Smart Systems and Emerging Technologies (RISE)

[2]Department of Mathematics and Computer Science, University of Perugia

*Keywords:* Abstract Argumentation Frameworks, Probabilistic Abstract Argumentation Frameworks, Normal Forms, Probabilistic Cliques, Probabilistic Logic Programming.

## 1. Introduction

Argumentation is an everyday method of humanity to discuss and solve myriad different situations where opinions or point of views conflict. Abstract Argumentation Frameworks [1] or Dung's Argumentation Frameworks (DAFs) aim in modeling everyday situations where information is inconsistent or incomplete. Many different extensions of DAFs from Dung's pioneering work have appeared in order to describe different everyday situations. Sample works includes assumption based argumentation [2], extending DAFs with support [3], introducing labels [4]. Furthermore, several generalizations of DAFs have been studied such as, Argumentation Frameworks with Recursive Attacks [5] which model attacks that can attack other attacks, abstract dialectical frameworks [6] is another generalization that allows acceptance conditions to arguments. Other directions that have been investigated includes bipolarity in argumentation frameworks [7], argumentation frameworks with preferences [8] and formalisms that extend DAFs by introducing weights in elements of the DAF, such as [9, 10]. Such approaches are powerful tools to model voting systems, belief in arguments and argument strength. A very thorough study of such approaches can be found at [11]

Knowledge representation with the use of probabilistic information has been used in many areas of computer science. Probabilistic information, is a powerful medium to represent knowledge. Similarly, many researchers have extended DAFs by adding probabilistic information. These very prominent extensions of DAFs have been categorized in two big groups by Hunter [12]: the **epistemic** approaches and the **constellation** approaches.

The epistemic approaches, such as those presented in [13, 14] describe probabilistic DAFs that the uncertainty does not alter the structure of the DAFs.

Furthermore, the epistemic approaches quantify the existing uncertainty (either of arguments being part of extensions, or argument label) instead of introducing new uncertainty.

The constellation approaches, such as those presented in [15, 16, 17, 18, 19] introduce probabilistic elements in the DAF in such a way that the structure of the DAF becomes uncertain. In difference from most constellation approaches, in [20] the authors define a constellation of DAFs by assigning probabilities directly to the graphs. The constellation approaches generate a set of DAFs with a probabilistic distribution and as such define a probabilistic distribution over the extensions of those DAFs.

In this paper we focus on the constellation approach from Li et al. [15]. [15] introduced probabilistic elements to the structure of DAFs, resulting to a set of DAFs. This allows for a set of arguments to be an (admissible, stable, ground, etc.) extension in some of the DAFs that are represented by the constellation. This simple but yet powerful representation has the ability to represent naturally many different uncertain scenarios.

The works of [21, 15] can be considered as the pioneering work on combining probabilities with DAFs. In this paper, we (a) define the probabilistic attack normal form for PrAAFs; (b) show how the normal form can represent any general PrAAF; and (c) illustrate the connection of the constellation semantics with probabilistic logic programming. We also (d) define and prove the same properties for the probabilistic argument normal form for PrAAFs and finally, (e) present a new probabilistic element for constellation semantics, namely probabilistic cliques.

The rest of the paper is structured as follows. First, we briefly introduce DAFs and PrAAFs. We then present the possible worlds notion, the probabilistic attack normal form for PrAAFs and a transformation of general PrAAFs to probabilistic attack normal form. We continue, by demonstrating the relation of PrAAFs with probabilistic logic programming and provide an implementation. Finally, we conclude and present future work.

## 2. Preliminaries

### *2.1. Abstract Argumentation*

Dung [1], introduced abstract argumentation frameworks (DAFs) which have been extended by many researchers.

**Definition 1.** An abstract argumentation frameworks is a tuple $DAF = (Args, Atts)$ where $Args$ is a set of arguments and $Atts$ a set of attacks among arguments of the form of a binary relation $Atts \subseteq Args \times Args$.

- For arguments $a, b \in Args$, we use $a \rightarrow b$ as a shorthand to indicate $(a, b) \in Atts$ and we say that argument $a$ *attacks* argument $b$.
- We say an argument $b$ is *defended* by a set $S \subseteq Args \iff \forall a \in Args$, if $a \rightarrow b$ then $\exists c \in S, c \rightarrow a$.

Figure 1 illustrates an example DAF and the notions of attack and defense in DAFs.

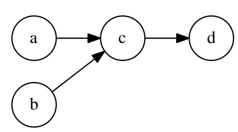



Figure 1: Example DAF $(\{a, b, c, d\}, \{a \rightarrow c, b \rightarrow c, c \rightarrow d\})$. Arguments are represented as cycles and attacks as arrows. Arguments $a$, $b$ are attacking argument $c$ which attacks argument $d$. We can also say that the set $\{a, d\}$ defends argument $d$ from the attack $c \rightarrow d$.

Given a DAF, and according to certain evaluation criteria (called semantics), the most common computational task in DAFs is to identify sets of arguments (called extensions) that yield by the semantics. Two important notions for the definitions of various kinds of extensions are conflict-free sets and acceptability of arguments. A set of arguments $S \subseteq Args$ is said to be *conflict-free* iff $\nexists a, b \in S$ where $a \rightarrow b \in Atts$. An argument $a \in Args$ is acceptable with respect to set

$S \subseteq Args$ if no argument attack $a$ or if $\forall b \in Args$ that $\exists b \rightarrow a \in Atts$ then $\exists c \in S$ where $c \rightarrow b \in Atts$.

Given the above [1] gives semantics to DAF by the use of extensions over subsets of arguments. Dung first defines the *admissible* semantics.

- A set $S \subseteq Args$ is admissible $\iff$ $S$ is conflict free and each $a \in S$ is acceptable with respect to $S$.
- A set $S \subseteq Args$ is preferred $\iff$ $S$ is a maximal (with respect to set inclusion) admissible set.
- A set $S \subseteq Args$ is complete $\iff$ $S$ is admissible and each argument that is acceptable with respect to $S$ is in $S$.
- A set $S \subseteq Args$ is grounded $\iff$ $S$ is the minimal (with respect to set inclusion) complete extension.

Following our example DAF from Figure 1, the set $\{a, b, d\}$ is admissible it is also preferred, complete and grounded. Over time several different semantics have been discussed such as stable [1], semi-stable [22], CF2 [23] etc. For further reading on DAFs semantics we direct the reader to [24].

### *2.2. Constellation based Probabilistic Abstract Argumentation Frameworks*

Hunter [21], categorizes probabilistic abstract argumentation frameworks (PrAAFs) in two different categories: the *constellation* and the *epistemic* PrAAFs. For this paper we will focus on the constellation approaches and we base our work in the definition of PrAAFs by [15].

A constellation approach to PrAAFs defines probabilities over the structure of the DAF graph. One can assign probabilities to either the arguments or/and attacks of the DAF. We refer to arguments/attacks with assigned probabilities less than 1 as probabilistic arguments/attacks and we refer as probabilistic elements to either probabilistic arguments or probabilistic attacks.

A probabilistic element $e$ exists in a DAF with probability $P(e)$. These probabilistic elements correspond to random variables, which are assumed to be mutually independent[3]. As such, a PrAAF defines a probability distribution over a set of DAFs.

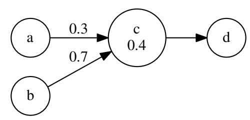



Figure 2: Example PrAAF $(\{a, b, c, d\}, \{1, 1, 0.4, 1\}, \{a \to c, b \to c, c \to d\}, \{0.3, 0.7, 1\})$. Arguments are represented as cycles and attacks as arrows. The probability of each element (unless if it equals with 1) appears over the attack or in the argument. Arguments a, b attack with a likelihood c which is a valid (exists) with a likelihood and attacks argument d.

**Definition 2.** Formally, a PrAAF is a tuple $PrAAF = (Args, P_{Args}, Atts, P_{Atts})$ where $Args$, $Atts$ define a DAF, $P_{Args}$ is a function mapping a probability for each $a \in Args$ with $0 < P_{Args}(a) \leq 1$ and $P_{Atts}$ is a function mapping a probability for each $\to \in Atts$ with $0 < P_{Atts}(\to) \leq 1$.

We note that for the remainder of the paper probabilities are notated as rational numbers in $(0, 1]$. Finally, stating an argument or an attack having probability 0 is redundant. A probabilistic argument or attack with 0 probability is an argument or attack that is not part of any DAF that the constellation represents. Figure 2, illustrates an example PrAAF with 3 different probabilistic elements.

### *2.3. Inducing DAFs by Imposing Restrictions*

Li et al. [15], restricted the combinations of probabilistic elements of PrAAFs to only those that generate valid DAFs. In order to successfully restrict the combinations, they introduced extra restrictions and also stated that the probabilities $P_{atts}$ are conditional probabilities instead the likelihood of existence for the attack. These restrictions appear in [15] as a separate definition, formally:

[3]As we are going to present later in the paper, the structure of DAF might impose dependencies among otherwise assumed independent probabilistic elements.

**Definition 3** (Inducing a DAF from a PrAAF)**.** A DAF $(Args_{Ind}, Atts_{Ind})$ is said to be induced from a PrAAF $(Args, P_{Args}, Atts, P_{Atts})$ iff all of the following hold:

1. $Args_{Ind} \subseteq Args$
2. $Atts_{Ind} \subseteq Atts \cap (Args_{Ind} \times Args_{Ind})$
3. $\forall a \in Args$ such that $P_{Args}(a) = 1, a \in Args_{Ind}$
4. $\forall a1 \to a2 \in Atts$ such that $P_{Atts}(a1 \to a2) = 1$ and $P_{Args}(a1) = P_{Args}(a2) = 1, a1 \to a2 \in Atts_{Ind}$

Furthermore, $P_{Att}(a1 \to a2)$ is stated to be the conditional probability of the attack existing when both attacking and attacked argument exist in the DAF $(P_{Att}(a1 \to a2 | a1, a2 \in Args_{Ind}))$.

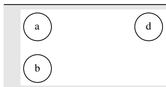
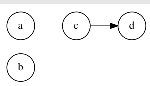
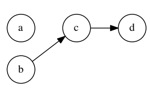
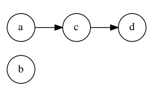
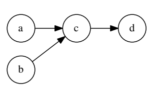
| **DAF** | **Possible World** | **Prob.** | **Admissible Sets** |
|---|---|---|---|
|  | $(\neg a \to c \wedge \neg b \to c \wedge \neg c) \vee$ $(\neg a \to c \wedge b \to c \wedge \neg c) \vee$ $(a \to c \wedge \neg b \to c \wedge \neg c) \vee$ $(a \to c \wedge b \to c \wedge \neg c)$ | 0.6 | $\{\}$, $\{a\}$, $\{b\}$, $\{d\}$, $\{a,b\}$, $\{a,d\}$, $\{b,d\}$, $\{a,b,d\}$ |
|  | $\neg a \to c \wedge \neg b \to c \wedge c$ | 0.084 | $\{\}$, $\{a\}$, $\{b\}$, $\{c\}$, $\{a,b\}$, $\{a,c\}$, $\{b,c\}$, $\{a,b,c\}$ |
|  | $\neg a \to c \wedge b \to c \wedge c$ | 0.196 | $\{\}$, $\{a\}$, $\{b\}$, $\{a,b\}$, $\{b,d\}$, $\{a,b,d\}$ |
|  | $a \to c \wedge \neg b \to c \wedge c$ | 0.036 | $\{\}$, $\{a\}$, $\{b\}$, $\{a,b\}$, $\{a,d\}$, $\{a,b,d\}$ |
|  | $a \to c \wedge b \to c \wedge c$ | 0.084 | $\{\}$, $\{a\}$, $\{b\}$, $\{a,b\}$, $\{a,d\}$, $\{b,d\}$, $\{a,b,d\}$ |

Table 1: Induced DAF of our example PrAAF from Figure 2. Shaded rows, illustrate an Induced DAF that contains multiple possible worlds that would generate an invalid DAF. With arrows in the possible world column we denote attacks and not implication.

Table 1 presents the induced DAFs from our example PrAAF[4]. Clearly, there is an exponential number of induced DAFs that a PrAAF represents. We find that the imposed restrictions from Li et al. [15] create a more complex and less intuitive PrAAF definition than what is necessary. By imposing these extra rules, one cannot handle probabilistic arguments and probabilistic attacks in the same way[5]. Furthermore, you cannot consider that probabilistic elements are (binary) random variables that generate combinations. While, these restrictions cause no theoretical problem, the require extra work when analyzing them and special handling when implemented.

## 3. Possible Worlds and DAFs

As mentioned a PrAAF defines a probability distribution for all the possible non-probabilistic DAFs it contains. Each single possible set of probabilistic elements (arguments or attacks) of the PrAAF can be called a **possible world**. Table 2 presents all possible worlds for the example PrAAF of Figure 2. One can notice that having only three different probabilistic elements it generates eight possible worlds. The possible worlds of a PrAAF are exponential in the number of probabilistic elements ($2^N$ where $N$ the number of probabilistic elements).

**Definition 4** (Probability of Possible World)**.** The probability of a possible world equals to the product of the probability of each probabilistic element that is in the possible world with the product of one minus the probability of each probabilistic element that is excluded from the possible world.

$$P_{world} = \prod_{e_i \in DAF_{world}} P(e_i) \cdot \prod_{e_j \notin DAF_{world}} (1 - P(e_j))$$

[4]For now we ask the reader to ignore the possible world column which is used later in the paper.

[5]In Li et al. [15] is claimed that the probability of an induced DAF is the joint probability of the independent probabilistic elements. But as probabilistic attacks are not completely independent the probability of the induced DAF requires the sum (= 1) of the depended probabilistic attacks when at least one of their connected arguments is not present. Doder and Woltran addressed this issue theoretically at [17] at definition 10.

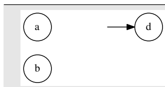
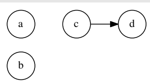
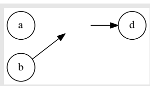
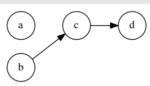
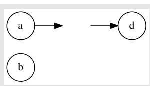
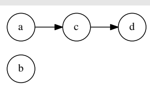
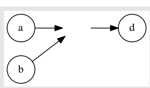
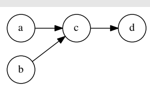
| DAF | Possible World | Prob. | Admissible Sets |
|---|---|---|---|
|  | $\neg a \rightarrow c \wedge \neg b \rightarrow c \wedge \neg c$ | 0.126 | $\{\}$, $\{a\}$, $\{b\}$, $\{d\}$, $\{a,b\}$, $\{a,d\}$, $\{b,d\}$, $\{a,b,d\}$ |
|  | $\neg a \rightarrow c \wedge \neg b \rightarrow c \wedge c$ | 0.084 | $\{\}$, $\{a\}$, $\{b\}$, $\{c\}$, $\{a,b\}$, $\{a,c\}$, $\{b,c\}$, $\{a,b,c\}$ |
|  | $\neg a \rightarrow c \wedge b \rightarrow c \wedge \neg c$ | 0.294 | $\{\}$, $\{a\}$, $\{b\}$, $\{d\}$, $\{a,b\}$, $\{a,d\}$, $\{b,d\}$, $\{a,b,d\}$ |
|  | $\neg a \rightarrow c \wedge b \rightarrow c \wedge c$ | 0.196 | $\{\}$, $\{a\}$, $\{b\}$, $\{a,b\}$, $\{b,d\}$, $\{a,b,d\}$ |
|  | $a \rightarrow c \wedge \neg b \rightarrow c \wedge \neg c$ | 0.054 | $\{\}$, $\{a\}$, $\{b\}$, $\{d\}$, $\{a,b\}$, $\{a,d\}$, $\{b,d\}$, $\{a,b,d\}$ |
|  | $a \rightarrow c \wedge \neg b \rightarrow c \wedge c$ | 0.036 | $\{\}$, $\{a\}$, $\{b\}$, $\{a,b\}$, $\{a,d\}$, $\{a,b,d\}$ |
|  | $a \rightarrow c \wedge b \rightarrow c \wedge \neg c$ | 0.126 | $\{\}$, $\{a\}$, $\{b\}$, $\{d\}$, $\{a,b\}$, $\{a,d\}$, $\{b,d\}$, $\{a,b,d\}$ |
|  | $a \rightarrow c \wedge b \rightarrow c \wedge c$ | 0.084 | $\{\}$, $\{a\}$, $\{b\}$, $\{a,b\}$, $\{a,d\}$, $\{b,d\}$, $\{a,b,d\}$ |

Table 2: Possible worlds of our example PrAAF from Figure 2. Shaded rows, illustrate possible worlds that generate an invalid DAF. With arrows in the possible world column we denote attacks and not implication.

While it is natural to use the notion of possible worlds in order to describe PrAAFs, unfortunately in general PrAAFs[6] not all possible worlds generate a valid DAF. Ideally, we want each possible world to generate a single unique valid DAF.

A second pitfall for general PrAAFs, lies in the combination of the independence assumption of PrAAFs probabilistic elements. We earlier stated that we assume each probabilistic element is independent from the other probabilistic elements. When a probabilistic argument is connected with a probabilistic attack and we consider a possible world where the probabilistic argument does not exist we are implicitly also forcing the probabilistic attack not to exist, thus creating an implicit dependency among probabilistic elements.

To better illustrate this pitfall of general PrAAFs, we use the example PrAAF of Figure 2. Consider the attack $a \to c$ which has a 0.3 probability of existence for each generated DAFs. If you sum the possible worlds of Table 2 where the edge $a \to c$ exists you do get a probability of 0.3 but if you sum the possible worlds of Table 2 where the edge $a \to c$ exists and it is a valid DAF then you get a probability of $0.036 + 0.084 = 0.12$ as it only exists in two possible worlds (rows 6 and 8 of Table 2) instead of the expected four possible worlds (rows 5 to 8 of Table 2). In other words, the probability of the attack existing in a valid DAF is not equal with the stated at the PrAAF and depends from the existence probability of arguments. For our example, it depends from argument $c$ with $P(c) = 0.4$ and $P(a \to c \in inducedDAF) = P(c) \cdot P(a \to c) = 0.3 \cdot 0.4 = 0.12$.

Notice at Table 1 that the highlighted induced DAF represents the four worlds that argument c is not part of the DAF. Also notice that the specific induced DAF is not part of any possible world, but is the corrected DAF of the four non valid DAFs that the possible worlds generate.

Partially motivated from the extensive research on how to compute the prob-

[6]We refer to general PrAAFs, as any constellation PrAAF that uses a definition similar to Definition 2. With the term general PrAAFs we do not include any extra restrictions imposed to the PrAAF definition.

abilities without explicitly enumerating all possible worlds, we illustrate how to transform any general PrAAF to a PrAAF that its Induced DAFs coincide with the possible worlds view. We point out some works from Probabilistic Logic Programming that is closely related with PrAAFs. For efficient exact inference we direct the reader at [25, 26, 27, 28, 29], and for efficient approximate inference at [30, 31, 32].

## 4. Probabilistic Attack Normal Form

In this section we introduce the Probabilistic Attack Normal Form for PrAAFs. The normal formed PrAAFs definition does not require any added restrictions in order for the PrAAFs to generate possible worlds with only valid DAFs. Also the probabilistic elements of normal formed PrAAFs are mutually independent and the probabilities of probabilistic elements always represents their likelihood of existence. These characteristics of normal formed PrAAFs allow for easier reasoning and also define a clearer probabilistic distribution.

**Definition 5** (Probabilistic Attack Normal Form)**.** A PrAAF $P$ is in its probabilistic attack normal form if it contains no probabilistic arguments ($\forall a \in Args, P(a) = 1$).

The probabilistic attack normal form definition does not fall to the aforementioned pitfalls and does not require further restrictions like the general PrAAFs definition. The pitfalls in general PrAAFs originate in the interaction of connected probabilistic arguments with probabilistic attacks. By having only probabilistic attacks the two pitfalls do not appear. Furthermore, the probabilistic attack normal form definition for PrAAFs is simpler and allows easier reasoning about PrAAF properties.

Finally, we are going to illustrate that having PrAAFs in the probabilistic attack normal form does not reduce the representation power of PrAAFs and that any probabilistic distribution that can be represented in general PrAAFs it can also be represented in the probabilistic attack normal form for PrAAFs.

Later on we also illustrate the same properties for the Probabilistic Argument Normal Form PrAAFs.

### *4.1. Transforming General PrAAFs to Probabilistic Attack Normal Form PrAAFs*

In this section we present a transformation that illustrates that any general PrAAF can be transformed to a Probabilistic Attack Normal Form. Both the original and the transformed PrAAF have the same probabilistic distribution over their extensions. Because of the existence of such a transformation one could use PrAAFs with only Probabilistic Attacks in order to represent any general PrAAF. Or, from a different perspective, one could define Probabilistic Arguments as syntactic sugar using Probabilistic Attacks and definite Arguments.

### *4.2. Transforming Probabilistic Arguments to Probabilistic Attacks*

Before we present the transformation of probabilistic argument to probabilistic attack, we need to define a special argument that we call *ground truth*[7]:

**Definition 6** (Ground Truth[8])**.** We introduce a special argument called *Ground Truth* and shorthand it with the letter $\eta$. We say that $\eta$ is undeniably true meaning that $\eta$ is never attacked by any argument and is always included in all extensions regardless the semantics used.

The $\eta$ argument modifies the extensions of a DAF for all semantics in such a way that $\eta$ must always be included. For example, in the admissible semantics of a DAF without $\eta$ a valid extension is the empty set($\{\}$), but in a DAF that contains $\eta$ the empty set is not a valid extension under the admissible semantics and the equivalent extension to the empty set is $\{\eta\}$. Note, that the extensions

[7] We use the same notation for ground truth ($\eta$) as in [3], we note though that the uses of ground truth in the two papers is different.

[8] The $\eta$ argument is only a construct we use in order to illustrate how Probabilistic Arguments can be transformed to Probabilistic Attacks and the PrAAF to retain the same probabilistic distribution.

of the original DAF and the extensions of a DAF with $\{\eta\}$ have a one-on-one correspondence for all semantics.

**Definition 7** (Acceptable Extensions)**.** For DAFs that contain $\eta$, an acceptable extension $E$ is one that includes $\eta$ ($\eta \in E$).

By using $\{\eta\}$ now we can define a transformation for general PrAAFs to Probabilistic Attack Normal Form as follows.

**Transformation 1** (General PrAAF to Probabilistic Attack Normal Form)**.** Any PrAAF $P$, can be transformed to an equivalent PrAAF $P'$ by removing any probabilistic information attached to an argument $a \in Args$, with $P(a)$ and introducing a probabilistic attack from the ground truth $\eta$ to argument $a$ with probability $1 - P(a)$.

Important note, $\eta$ is a construct we use in order to illustrate that any general PrAAF can be transformed to a Probabilistic Attack Normal Form PrAAF. When someone would define their own PrAAFs the $\eta$ construct would not appear to the user as: 1) you either do not use probabilistic arguments; 2) if you use the would be automatically converted by syntactic sugar to probabilistic attacks and a hidden to the user $\eta$. The semantics of PrAAFs and the probabilistic distribution would remain identical for the user but for the underline algorithm it would be required to only evaluate probabilistic attacks instead both probabilistic attacks and probabilistic arguments.

**Definition 8.** We notate $P \equiv^{\sigma}_{|\eta} P'$ the standard equivalence [33] of PrAAF $P$ with PrAAF $P'$ under semantics $\sigma$ by ignoring the existence of $\eta$ in the acceptable extensions.

**Theorem 1** (Equivalence of transformed PrAAF)**.** *The admissible extensions $E'$ containing $\eta$ of the transformed PrAAF $P'$ and the admissible extensions $E$ of the original PrAAF $P$ follow the same (equivalent) probabilistic distribution.*

*Proof.* We split the proof in two parts. First we show that PrAAF $P$ generates DAFs that have the same admissible sets with the generated DAFs from PrAAF

$P'$. We point out that for PrAAF $P'$ acceptable admissible sets are only the ones that contain the ground truth argument which we ignore its existence when comparing admissible sets. For example, the empty admissible set of $P$ is equivalent with the $\{\eta\}$ admissible set of $P'$. A probabilistic argument $pa$ generates two different sets of DAFs, set $S_1$ where $pa$ exists and $S_2$ where $pa$ does not exist.

PrAAF $P'$ generates $S'_1$ the equivalent sets of $S_1$ when $\eta \to pa$ does not exist and the equivalent $S'_2$ sets of $S_2$ when $\eta \to pa$ exists. When comparing a DAF with $pa$ versus a DAF without $\eta \to pa$ the only difference is the existence of $\eta$ as we only consider admissible sets that contain it and we ignore its existence in the admissible sets the two graphs are equivalent thus the $S'_1$ sets are equivalent with the $S_1$ sets.

For $S_2$ where $pa$ does not exist, the equivalent $S'_2$ contains DAFs where the argument $pa$ is been attacked by $\eta$ and is not defended by any other argument. Clearly, as $\eta$ is included in every extension that we consider then every attack originating from $pa$ is defended; thus, the DAFs of $S_2$ generate the same admissible extensions with the DAFs of $S'_2$.

Next part is to show that the probability of each extension is the same. The probability that a set is an admissible extension is been computed by the summation of the possible worlds where that set is admissible. As $S_1, S_2$ are equivalent with $S'_1, S'_2$ and produce equivalent DAFs then the possible worlds are equivalent too. The probability of each possible world is also the same as when $pa$ would exist the possible world probability is multiplied by $P(pa)$. In the equivalent case the attack $\eta \to pa$ does not exist and the possible world probability is multiplied by $1 - (1 - P(pa)) = pa$. Similarly, for the possible worlds that $pa$ does not exist. $\square$

**Corollary.** *PrAAF $P'$ has equivalent acceptable extensions with PrAAF $P$ for all semantics where acceptability of an argument is necessary for the inclusion of the argument in the extension. Such semantics include: complete, preferred, ground and stable semantics. Similarly, as the probabilistic distributions are* *equivalent then all acceptable extensions of $P'$ will have equal probability with their equivalent extension from $P$.*

We also want to point out that each general PrAAF can be transformed to a unique Probabilistic Attack Normal Form containing $\eta$. Also any Probabilistic Attack Normal Form that contains $\eta$ is reversible to the general PrAAF. For those reasons we can claim that the transformation is a one-to-one reversible transformation.

**Proposition 1** (Reversibility of the transformation)**.** *The general PrAAG to Probabilistic Attack Normal Form transformation is reversible and creates a one-on-one equivalent PrAAF.*

*Proof.* Any argument $a$ that is attacked by $\eta$ is transformed to a Probabilistic Argument with $(1 - P(\eta \to a))$ probability. Finally, one can drop $\eta$ to return to the original general PrAAF. $\square$

By using the general PrAAF to Probabilistic Attack Normal Form transformation to the PrAAF of Figure 2 we get the PrAAF of Figure 3. Table 3 presents the possible worlds of the PrAAF of Figure 3. Now, each possible world represents a valid DAF that generates the equivalent acceptable admissible sets like the original PrAAF. Furthermore, the probabilistic distribution is identical.

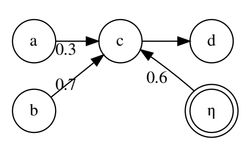



Figure 3: Example transformed PrAAF $(\{a, b, c, d, \eta\}, \{1, 1, 1, 1, 1\}, \{a \to c, b \to c, c \to d, \eta \to c\}, \{0.3, 0.7, 1, 0.6\})$.

**Proposition 2** (Complexity of the Transformation)**.** *The general PrAAF to Probabilistic Attack Normal Form transformation has linear complexity $O(N)$*

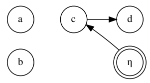
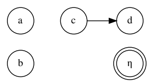
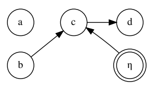
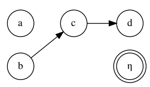
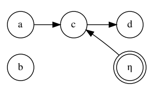
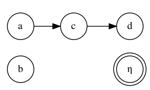
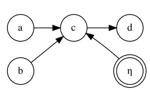
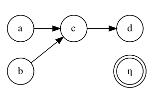
| DAF | Possible World | Prob. | Acceptable Admissible Sets |
|---|---|---|---|
|  | $\neg a \rightarrow c \wedge \neg b \rightarrow c \wedge \eta \rightarrow c \equiv \neg a \rightarrow c \wedge \neg b \rightarrow c \wedge \neg c$ | 0.126 | $\{\eta\}, \{\eta, a\}, \{\eta, b\}, \{\eta, d\}, \{\eta, a, b\}, \{\eta, a, d\}, \{\eta, b, d\}, \{\eta, a, b, d\}$ |
|  | $\neg a \rightarrow c \wedge \neg b \rightarrow c \wedge \neg\eta \rightarrow c \equiv \neg a \rightarrow c \wedge \neg b \rightarrow c \wedge c$ | 0.084 | $\{\eta\}, \{\eta, a\}, \{\eta, b\}, \{\eta, c\}, \{\eta, a, b\}, \{\eta, a, c\}, \{\eta, b, c\}, \{\eta, a, b, c\}$ |
|  | $\neg a \rightarrow c \wedge b \rightarrow c \wedge \eta \rightarrow c \equiv \neg a \rightarrow c \wedge b \rightarrow c \wedge \neg c$ | 0.294 | $\{\eta\}, \{\eta, a\}, \{\eta, b\}, \{\eta, d\}, \{\eta, a, b\}, \{\eta, a, d\}, \{\eta, b, d\}, \{\eta, a, b, d\}$ |
|  | $\neg a \rightarrow c \wedge b \rightarrow c \wedge \neg\eta \rightarrow c \equiv \neg a \rightarrow c \wedge b \rightarrow c \wedge c$ | 0.196 | $\{\eta\}, \{\eta, a\}, \{\eta, b\}, \{\eta, a, b\}, \{\eta, b, d\}, \{\eta, a, b, d\}$ |
|  | $a \rightarrow c \wedge \neg b \rightarrow c \wedge \eta \rightarrow c \equiv a \rightarrow c \wedge \neg b \rightarrow c \wedge \neg c$ | 0.054 | $\{\eta\}, \{\eta, a\}, \{\eta, b\}, \{\eta, d\}, \{\eta, a, b\}, \{\eta, a, d\}, \{\eta, b, d\}, \{\eta, a, b, d\}$ |
|  | $a \rightarrow c \wedge \neg b \rightarrow c \wedge \neg\eta \rightarrow c \equiv a \rightarrow c \wedge \neg b \rightarrow c \wedge c$ | 0.036 | $\{\eta\}, \{\eta, a\}, \{\eta, b\}, \{\eta, a, b\}, \{\eta, a, d\}, \{\eta, a, b, d\}$ |
|  | $a \rightarrow c \wedge b \rightarrow c \wedge \eta \rightarrow c \equiv a \rightarrow c \wedge b \rightarrow c \wedge \neg c$ | 0.126 | $\{\eta\}, \{\eta, a\}, \{\eta, b\}, \{\eta, d\}, \{\eta, a, b\}, \{\eta, a, d\}, \{\eta, b, d\}, \{\eta, a, b, d\}$ |
|  | $a \rightarrow c \wedge b \rightarrow c \wedge \neg\eta \rightarrow c \equiv a \rightarrow c \wedge b \rightarrow c \wedge c$ | 0.084 | $\{\eta\}, \{\eta, a\}, \{\eta, b\}, \{\eta, a, b\}, \{\eta, a, d\}, \{\eta, b, d\}, \{\eta, a, b, d\}$ |

Table 3: Possible worlds after transforming PrAAF of Figure 2. With arrows in the possible world column we denote attacks and not implication.

*to the number of probabilistic arguments $N$ that the original PrAAF contains. It grows the size of the original PrAAF by one argument and by $N$ attacks. The transformation does not affect the worst case complexity of computing any extension or the probability that a set is any type of an extension.*

### *4.3. A Short Discussion on $\eta$*

Baring in mind that we have introduced the construct $\eta$ we want to have a short discussion on its actual meaning. There is three major discussion points relating $\eta$.

1. $\eta$ is used as a construct in order to define a transformation from probabilistic argument to probabilistic attack and thus provide a tool to define probabilistic arguments as syntactic sugar in any implementation.
2. With the transformation that uses $\eta$ its proven that any PrAAF has an equivalent PrAAF in probabilistic attack normal form. As a result one can use PrAAFs only with probabilistic attacks. Thus, all formal work can be restricted to the simpler PrAAFs and proofs can be simplified to only those.
3. Finally, we are giving $\eta$ the meaning of ground truth, an argument that is not being attacked and is always believed (and for that reason it replaces the empty set in all extensions).

We do not want to confuse the reader thinking that $\eta$ must appear in PrAAFs, the opposite actually. We use $\eta$ only when we require to transform a PrAAF that contains a probabilistic argument to one that does not contain any. Our suggestion is to only use probabilistic attacks and not probabilistic arguments as you get equal expressible and simpler PrAAFs.

## 5. Implementing PrAAFs with Probabilistic Logic Programming

In Probabilistic Logic Programming (PLP), a probabilistic distribution is defined by introducing probabilities over logic clauses or logic facts. The Distribution Semantics [34] formalized the meaning of such probabilities and clarified that each possible world is a logic program. Later the work presented in [35] illustrated that attaching probabilities in different ways can be transformed in equivalent Probabilistic Logic programs under the different formulations that used the Distribution Semantics.

With our work we illustrated that general PrAAFs can be transformed to generate possible worlds instead of induced DAFs while maintaining the same probabilistic distribution. As a result, one can convert any PrAAF to a Probabilistic Logic program that has an equivalent probabilistic distribution.

In order to implement a solver for PrAAFs we used *MetaProbLog* [36][9] and the web interface of ConArg [37]. MetaProbLog is a PLP framework based on ProbLog [27] semantics. The PrAAF implementation first appeared in [38] and is publicly available[10].

### 5.1. ProbLog

A ProbLog program $T$ [27] consists of a set of facts annotated with probabilities $p_i :: pf_i$ – called *probabilistic facts* – together with a set of standard definite clauses $h : -b_1, \ldots, b_n$. that can have positive and negative probabilistic literals in their body. A probabilistic fact $pf_i$ is true with probability $p_i$. These facts correspond to random variables, which are assumed to be mutually independent. Together, they thus define a distribution over subsets of $L_T = \{pf_1, \ldots, pf_n\}$. The definite clauses add arbitrary *background knowledge* (BK) to those sets of *logical* facts. To keep a natural interpretation of a ProbLog program we assume that probabilistic facts cannot unify with other probabilistic facts or with the background knowledge rule heads.

**Definition 9.** ProbLog Program: Formally, a ProbLog program is of the form $T = \{pf_1, \ldots, pf_n\} \cup BK$.

Given the one-to-one mapping between ground definite clause programs and Herbrand interpretations, a ProbLog program defines a distribution over its

[9] MetaProbLog is available at: `www.dcc.fc.up.pt/metaproblog`
[10] `http://www.dmi.unipg.it/conarg/`

Herbrand interpretations.

The distribution semantics are defined by generalising the least Herbrand models of the clauses by including subsets of the probabilistic facts. If fact $pf_i$ is annotated with $p_i$, $pf_i$ is included in a generalised least Herband model with probability $p_i$ and left out with probability $1 - p_i$. The different facts are assumed to be probabilistically independent, however, negative probabilistic facts in clause bodies allow the user to enforce a choice between two clauses.

As such, a ProbLog program specifies a probability distribution over all its possible non-probabilistic subprograms. The success probability of a query is defined as the probability that the query succeeds in such a random subprogram. ProbLog follows the distribution semantics [34].

### *5.2. PrAAF to ProbLog Transformation*

We automatically transform any PrAAF in probabilistic attack normal form described by the ConArg web interface to a MetaProbLog program. For example the PrAAF of Figure 3 generates the following MetaProbLog Program:[11].

```
arg(a).
arg(b).
arg(c).
arg(d).
arg(eta).
0.3::att(a,c).
0.7::att(b,c).
1.0::att(c,d).
0.6::att(eta,c).
```

Then as Background knowledge we can introduce any semantics of Argumentation Frameworks as MetaProbLog predicates like in Figure 4. As we are going

[11]The graphical interface annotates probabilistic attacks as `att(a,c):-0.3.` and probabilities of 1.0 can be omitted.

to use dynamic programming for our computations cyclic handling is left for the MetaProbLog engine.

Finally, we need to define our query. As shown earlier a PrAAF describes an exponential number (to the probabilistic elements) of DAFs, and each DAF has its own extensions. For that reason we focus our queries on computing the probability that a set is an extension under some semantics. For example: `:- problog_exact(admissible([a,d,eta])).` and leave the computation to the MetaProbLog engine.

### *5.3. MetaProbLog Inference*

MetaProbLog provides several different efficient probabilistic inference methods such as: (i) exact inference based on Reduced Ordered Binary Decision Diagrams (ROBDDs) and dynamic programming [25]; (ii) program (DAF) sampling with memoization [39]; (iii) any-time inference using an iterative deepening algorithm [40].

For now the web interface exposes two forms of probabilistic inference: exact and DAF sampling. The exact inference method computes the exact probability; the DAF sampling inference method is an approximation method. In most cases the exact inference is able to compute the result faster than most approximation methods, such as the DAF sampling inference. But there exist cases where exact inference is intractable and a user is forced to use an approximation method, for those cases we provide the DAF sampling inference.

MetaProbLog's exact inference follows three steps to prove a query and calculate the probability of its success.

1. **SLD resolution** is used to prove the query and collect Boolean formulae that represent the possible worlds.
2. **Boolean formulae preprocessing** is used to optimize the collected Boolean formulae.
3. **ROBDD compilation** is used to compile all collected Boolean formulae in Reduced Order Binary Decision Diagram (ROBDD).

```
:- problog_table conflict_free/1.
:- problog_table admissible/1.
:- problog_table admissible/2.

attacked(A, B) :-
  att(B, A), arg(B).

conflict_free([]).
conflict_free([A|S]) :-
  member(X, [A|S]),
  problog_not(attacked(A, X)),
  problog_not(attacked(X, A)),
  conflict_free(S).

acceptable(A, _S) :-
  problog_not(attacked(A, _X)).
acceptable(A, S) :-
  attacked(A, B),
  member(C, S),
  attacked(B, C).

admissible([]).
admissible(S) :-
  conflict_free(S),
  admissible(S, S).

admissible([], _).
admissible([A|T], S) :-
  acceptable(A, S),
  admissible(T, S).
```

Figure 4: Conflict free and admissible definitions in MetaProbLog. `problog_table/1` directive activates SLG resolution for probabilistic programs, for more details we refer the reader to [25]. `problog_not/1` is implementing general negation for probabilistic logic programs, for more details we refer the reader to [36].

For a descritpion of the implementation of MetaProbLog's exact inference we address the reader to [25, 27]. The implementation depends heavily in dynamic programming in order to avoid recomputations and to provide efficient inference.

The program sampling inference is based on the use of Monte Carlo methods, that is, to use the ProbLog program to generate large numbers of random subprograms[12] and to use those to estimate the probability. More specifically, such a method proceeds by repeating the following steps:

1. sample a logic (sub)program $L$ from the ProbLog program
2. search for a proof of the initially stated query $q$ in the sample $L \cup BK$
3. estimate the success probability as the fraction $P$ of samples which hold a proof of the query

The implementation of this approach for MetaProbLog, is similar with the one described at [39], and takes advantage of the independence of probabilistic facts to generate samples lazily while proving the query, that is, sampling and searching for proofs which are interleaved. To assess the precision of the current estimate $P$, for each $N$ samples the width $\delta$ of the 95% confidence interval is approximated as

$$\delta = 2 \cdot \sqrt{\frac{P \cdot (1 - P)}{N}} \tag{1}$$

If the number of samples $N$ is large enough the interval of confidence becomes smaller, and the probability that the estimate is close to the true probability of the query increases.

## 6. Probabilistic Argument Normal Form

Similarly with the Probabilistic Attack Normal Form one could allow only probabilistic arguments for PrAAFs, defining a Probabilistic Argument Normal Form for PrAAFs. In this section we are going to illustrate that both normal forms are viable and equivalent. Still the probabilistic distribution of

[12] We note that, each sampled ProbLog program corresponds to sampling a DAF.

the Probabilistic Argument Normal Form introduces a bit of complexity. By (explicitly) defining a probability at arguments one also (implicitly) defines a probability at the connected attacks. For that reason one should extend the scope of the probability attached to an argument from the existence of the argument to the existence of both the argument and the attacks connected to that argument. Formally, if $P_{Args}(a)$ is the likelihood that $a \in Args_{Ind}$; then $P(a \rightarrow b)$ is the (implicit) likelihood that $a \rightarrow b \in Atts_{Ind}$ and is computed as $P(a \rightarrow b) = P_{Args}(a) \cdot P_{Args}(b)$; similarly for $P(b \rightarrow a)$.

**Definition 10** (Probabilistic Argument Normal Form)**.** A PrAAF $P$ is in its probabilistic argument normal form if it contains no probabilistic attacks ($\forall x \rightarrow y \in Atts, P(x \rightarrow y) = 1$).

Like with Probabilistic Attack Normal Form PrAAFs, also PrAAFs in Probabilistic Argument Normal Form have possible worlds that are valid DAFs. Furthermore, in a similar way with Probabilistic Attack Normal Form, one can illustrate that Probabilistic Argument Normal Form PrAAFs have equal representation power with general PrAAFs and that any probabilistic distribution that can be represented in general PrAAFs it can also be represented in Probabilistic Argument Normal Form PrAAFs.

### *6.1. Transforming Probabilistic Attacks to Probabilistic Arguments*

In this section, we present two transformations from Probabilistic Attack to Probabilistic Argument. The first transformation is simpler to apply and in most cases is the preferable transformation. The second transformation while slightly more complex in some cases produces smaller PrAAFs. Furthermore, it is interesting that there exist two different transformations for the second normal form.

**Transformation 2** (Probabilistic attack to probabilistic argument)**.** Any PrAAF $P$, can be transform to an equivalent PrAAF $P'$ by:

1. removing probabilistic attack $x \rightarrow y \in Atts$, with $0 < P(x \rightarrow y) < 1$

2. adding two auxiliary probabilistic arguments $xy1$, $xy2$ with probability $P(xy2) = P(x \to y)$
3. adding attacks $x \to xy1$, $xy1 \to xy2$ and $xy2 \to y$

**Theorem 2** (Equivalence of transformed PrAAF)**.** *The admissible extensions $E'$ not containing any auxiliary probabilistic arguments the transformed PrAAF $P'$ and the admissible extensions $E$ of the original PrAAF $P$ follow the same (equivalent) probabilistic distribution.*

*Proof.* In order to prove that the generated extensions from the transformed PrAAF $P'$ are equivalent with the original $P$ we break the proof in two parts. First step, as $P'$ maintained the same amount of probabilistic elements with the same probabilities the possible worlds that $P'$ generates is clearly equivalent with the same probability with the possible worlds that $P$ generates. Second step is to show that the extensions that are generated and selected from $P'$ are equivalent with the ones generated from $P$. Lets consider the possible worlds that the probabilistic attack $x \to y$ of the original PrAAF is present. These worlds are equivalent with the worlds of $P'$ when the probabilistic argument $xy$ is present. For $P'$, $x$ attacks the auxiliary argument $xy1$ which is the only argument that attacks the auxiliary argument $xy2$ thus $xy2$ would be defended whenever $x$ is defended, thus $xy2$ will attack $y$ whenever $x$ is defended which has the same effect on the extensions of $P'$ as the original $x \to y$ attack would have on the extensions of $P$. Similarly for, the case where the attack $x \to y$ would not be part of the possible worlds. Of course for getting the identical extensions we need to drop all extensions that include the auxiliary argument as the are redundant. We want to note, that not dropping the extensions that contain the auxiliary arguments does not affect the probabilities of the original extensions and can be used normally for probabilistic inference. □

For a short clarification in the probabilistic attack normal form transformation, the probability that a set is an extension over some semantics was the same with the probability that the union of the set and the ground truth is an extension of the same semantics. In the probabilistic argument normal form transformation the probability of a set being an extension of some semantics is the same, just extensions that include the auxiliary arguments also exist which can be ignored (for example when generating all possible extensions under some semantics).

This transformation grows significantly the DAF as it introduces two auxiliary arguments and three attacks for each probabilistic attack that is removed. Figure 5 presents the equivalent transformed PrAAF for our example PrAAF of Figure 2.

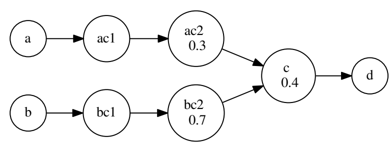



Figure 5: Example transformed PrAAF $(\{a, b, c, d, ac1, bc1, ac2, bc2\}, \{1, 1, 0.4, 1, 1, 1, 0.3, 0.7\}, \{a \rightarrow ac1, ac1 \rightarrow ac2, ac2 \rightarrow c, b \rightarrow bc1, bc1 \rightarrow bc2, bc2 \rightarrow c, c \rightarrow d\}, \{1, 1, 1, 1, 1, 1, 1\})$.

**Proposition 3** (Complexity of the Transformation)**.** *The probabilistic attack to probabilistic argument transformation has linear complexity $O(N)$ to the number of probabilistic attacks $N$. Similarly, the size of the PrAAF grows by $2 * N$ arguments and $2 * N$ attacks. Again, this transformation does not affect the worst case complexity of computing any extension or the probability that a set is an extension.*

Finally, we also present an alternative more complex transformation for a probabilistic attack to probabilistic argument that in some cases can be more compact.

**Transformation 3** (Probabilistic attack to probabilistic argument)**.** Any PrAAF $P$, can be transform to an equivalent PrAAF $P'$ by:

1. removing probabilistic attack $x \rightarrow y \in Atts$, with $0 < P(x \rightarrow y) < 1$
2. adding an auxiliary probabilistic argument $xy$ with probability $P(xy) = P(x \rightarrow y)$
3. adding attack $xy \rightarrow y$

4. $\forall z \to x \in Atts$ with $P(z \to x) = 1$ add an attack $z \to xy$.
5. $\forall yz \in Args$, where $yz$ are introduced auxiliary arguments, add an attack $xy \to yz$.

**Theorem 3** (Equivalence of transformed PrAAF)**.** *The admissible extensions $E'$ not containing any auxiliary probabilistic arguments the transformed PrAAF $P'$ and the admissible extensions $E$ of the original PrAAF $P$ follow the same (equivalent) probabilistic distribution.*

*Proof.* Similarly, with the previous theorem. □

Figure 6 presents the equivalent transformed PrAAF for our example PrAAF of Figure 2.

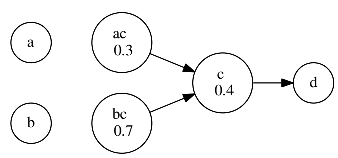



Figure 6: Example transformed PrAAF $(\{a, b, c, d, ac, bc\}, \{1, 1, 0.4, 1, 0.3, 0.7\}, \{ac \to c, bc \to c, c \to d\}, \{1, 1, 1\})$.

**Proposition 4** (Complexity of the Transformation)**.** *The probabilistic attack to probabilistic argument transformation has linear complexity $O(N+M)$ to the number of probabilistic attacks $N$ plus the number of incoming attacks $M$ that attack arguments that a probabilistic attack originates. Similarly, the size of the PrAAF grows by $N$ arguments and $M$ attacks (this transformation also removes $N$ original attacks and replaces them with $N$ attacks originating from auxiliary nodes). Again, this transformation does not affect the worst case complexity of computing any extension or the probability that a set is an extension.*

The second transformation produces a smaller graph for an argument that one or more probabilistic attack(s) originate when $4*N > N+M \to M < 3*N$ with $N$ the probabilistic attacks and $M$ the attacks directed to the originating probabilistic node. Both transformations could be used in order to produced the smallest transformed graph. For the rest of the paper we focus on the Probabilistic Attack Normal Form for PrAAFs from Definition 5.

## 7. Cliques in PrAAFs

Up until now, we have only discussed PrAAFs for which we assume independence among their probabilistic elements (attacks or arguments). In this section, we introduce a new form of probabilistic elements that (a) is compatible and may co-exist in a PrAAF with the existing probabilistic elements (probabilistic attacks in our case); (b) in a transparent way introduce dependencies among probabilistic element. We refer to these probabilistic elements as *Probabilistic Cliques*. Probabilistic cliques are an extension to the probabilistic semantics of PrAAFs.

But before we introduce probabilistic cliques lets start by defining what a clique is in a DAF.

**Definition 11** (Cliques in DAFs)**.** We call a *Clique* in DAF $\{Args, Atts\}$ a set of Arguments $C \subseteq Args$ that each argument in $C$ attacks each other. Formally: $\forall A, B \in C \exists A \to B \in Atts \wedge B \to A \in Atts$.

Cliques in DAFs are implicitly representing a mutual exclusive relation among the clique arguments over the extensions of the framework. Take for example the DAF of Figure 7, only one (or none) of the arguments $a, b, c$ can appear in any extension (in order to be conflict-free) of the DAF. This mutual exclusive relation, as illustrated in [21], imposes a problem to the assumption of probabilistic element independence of [15]. It is not in the context of this paper to re-discuss the distribution problems of cliques in general PrAAFs, neither the solution suggested by [21]. On the other hand we want to point out that our proposed probabilistic cliques does not have the problem discussed in [21].

### 7.1. Mutual Exclusive Probabilistic Arguments

In probabilistic reasoning [41], it is common to use mutually exclusive probabilistic elements to define different probabilistic logics; examples include [34, 42].

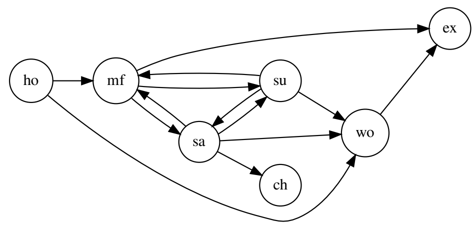



Figure 7: Example of a DAF $(\{ho, mf, sa, su, ch, wo, ex\}, \{ho \rightarrow mf, ho \rightarrow wo, mf \rightarrow sa, mf \rightarrow su, mf \rightarrow ex, sa \rightarrow mf, sa \rightarrow su, sa \rightarrow ch, sa \rightarrow wo, su \rightarrow mf, su \rightarrow sa, su \rightarrow wo, wo \rightarrow ex\})$ with a clique $(\{mf, sa, su\})$. Arguments are represented by cycles and attacks are represented by arrows. Cliques in DAFs have the characteristic that only one of their arguments can be a member of any admissible set. For this reason we can say that each argument of a clique is mutually exclusive to the rest arguments of the clique in relation to the admissible extensions.

Furthermore, other probabilistic logics are extended to support mutually exclusive probabilistic elements such as [43, 44]. These mutually exclusive probabilistic elements have found different applications and are used to realize Hidden Markov Models [45], Bayesian Networks and can be a crucial tool in expressing knowledge for probabilistic models.

We introduce the notion of Mutual Exclusive Probabilistic Arguments in PrAAFs as a set $PC$ of dependent probabilistic arguments, where $\sum_{\forall A \in PC} P(A) = 1$. We refer to each such set of mutual exclusive probabilistic arguments as a *Probabilistic Clique*. Furthermore, at each possible world exactly one and only one mutual exclusive probabilistic argument from each probabilistic clique can exist. More formally:

**Definition 12** (Probabilistic Clique & Mutual Exclusive Probabilistic Argument)**.** A *Probabilistic Clique* ($PC$), in PrAAFs, is a set of *Mutually Exclusive Probabilistic Arguments* ($A$) so that: $\sum_{\forall A \in PC} P(A) = 1$ and for each possible world $PW$, if $A \in PW, \forall B \in PC \wedge B \neq A \implies B \notin PW$.

The above definition extends the definition of possible worlds from Section 3 so that a possible world cannot contain two or more mutually exclusive probabilistic arguments. For set notation we denote a probabilistic cliques as a single argument that has the mutual exclusive sub arguments and we do not mention the attacks among each mutual exclusive argument. Figure 8 presents the Probabilistic clique $\{[mf; sa; su]\}$, $\{[0.7; 0.15; 0.15]\}$ in a PrAAF.

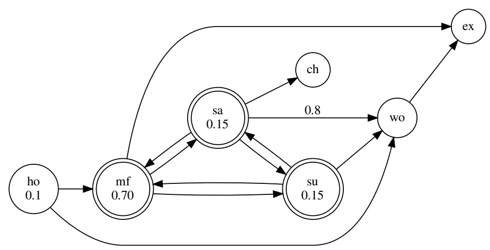



Figure 8: Example of a General PrAAF with a probabilistic clique $(\{ho, [mf; sa; su], ch, wo, ex\}, \{0.1, [0.7; 0.15; 0.15], 1, 1, 1\}, \{ho \rightarrow mf, ho \rightarrow wo, mf \rightarrow ex, sa \rightarrow ch, sa \rightarrow wo, su \rightarrow wo, wo \rightarrow ex\}, \{1, 1, 1, 1, 0.8, 1, 1\})$. Arguments are represented by cycles, arguments that are members of the probabilistic clique are notated with double circles, finally attacks are represented by arrows.

The use case behind probabilistic cliques, is to introduce knowledge that is known to be exclusive. For example, a probabilistic clique could contain as arguments what day of the week is like Monday-Friday, Saturday, Sunday enumerating all possible values. Each argument could be attacked or attack any other external argument. Like, *if it is Sunday then I was not at work*, etc. Probabilistic cliques enforce that only one such argument exists at all times describing a set of PrAAFs one for each clique argument. In order to describe a none of these arguments case an extra argument needs to be added in the probabilistic clique, such an argument could be implicitly defined by providing a distribution that sums less than 1.0 then the system would automatically define the none argument with the remaining probability.

Probabilistic cliques extend the semantics of general PrAAFs by introducing a third type of probabilistic element: probabilistic attack, probabilistic argument and probabilistic clique. Similarly, with general PrAAFs, probabilistic cliques require us to impose the same restrictions like in Definition 3 and adds the following restriction: $\forall PC \exists! A \in PC \wedge A \in Args_{Ind}, \forall B \in PC \wedge B \neq A \implies B \notin Args_{Ind}$. Probabilistic cliques share similarities with the mutual exclusive elements introduced in [46].

### 7.2. Probabilistic Cliques, Induced DAFs and Possible Worlds

Probabilistic cliques induce multiple (one for each argument in the clique) PrAAFs which in their own respect induce multiple DAFs. We use the general PrAAF example that uses all three different probabilistic elements in order to illustrate the induced DAFs and their respective possible worlds. Table 4, enumerates the induced DAFs of the general PrAAF with a probabilistic clique of Figure 8. Notice that the possible world column now contains a non binary random variable, the probabilistic clique.

### 7.3. Epistemic Properties of Probabilistic Cliques

Probabilistic cliques with dependent arguments (as the ones we introduced) introduce epistemic properties in constellation semantics. A clique (which is also a cyclic relation) introduces ambiguity in a DAF even before we extend the DAF with probabilistic elements. Probabilistic cliques quantify this ambiguity and for that reason the have an epistemic meaning. While it is not in the scope of this paper, it can be shown that PrAAFs that contain only probabilistic cliques have equivalent epistemic probabilistic DAFs as defined in [13, 14]. For an example we refer the reader to Examples 9 and 10 of [14]. The epistemic approach for probabilistic DAFs concluded that for the coherent semantics the probabilities of the two surveillance cameras must sum to 1, and "naturally" formed a probabilistic clique.

## 8. Conclusion and Future Work

In this paper, we (a) make the connection of induced DAFs with possible worlds; (b) formally introduce the Probabilistic Attack Normal Form for PrAAFs; (c) illustrate that the Probabilistic Attack Normal Form is sufficient to represent any general PrAAF as defined by [15]; (d) demonstrate that PrAAFs

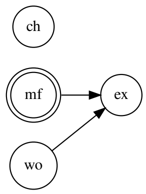
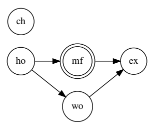
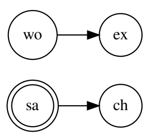
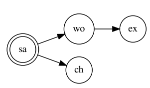
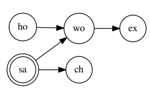
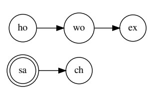
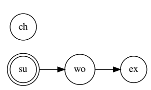
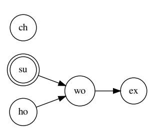
| DAF | Possible World | Prob. | Admissible Sets |
|---|---|---|---|
|   | $(PC = mf \wedge \neg ho \wedge \neg sa \rightarrow wo) \vee (PC = mf \wedge \neg ho \wedge sa \rightarrow wo)$ | 0.63 | $\{\}$, $\{ch\}$, $\{mf\}$, $\{wo\}$, $\{ch, mf\}$, $\{ch, wo\}$, $\{mf, ch\}$, $\{ch, mf, wo\}$ |
|   | $(PC = mf \wedge ho \wedge \neg sa \rightarrow wo) \vee (PC = mf \wedge ho \wedge sa \rightarrow wo)$ | 0.07 | $\{\}$, $\{ch\}$, $\{ho\}$, $\{ch, ho\}$, $\{ho, ex\}$, $\{ch, ho, ex\}$ |
|   | $PC = sa \wedge \neg ho \wedge \neg sa \rightarrow wo$ | 0.027 | $\{\}$, $\{wo\}$, $\{sa\}$, $\{wo, sa\}$ |
|   | $PC = sa \wedge \neg ho \wedge sa \rightarrow wo$ | 0.108 | $\{\}$, $\{sa\}$, $\{sa, ex\}$ |
|   | $PC = sa \wedge ho \wedge sa \rightarrow wo$ | 0.012 | $\{\}$, $\{ho\}$, $\{sa\}$, $\{ho, sa\}$, $\{ho, ex\}$, $\{ho, sa, ex\}$ |
|   | $PC = sa \wedge ho \wedge \neg sa \rightarrow wo$ | 0.003 | $\{\}$, $\{ho\}$, $\{sa\}$, $\{ho, sa\}$, $\{ho, ex\}$, $\{ho, sa, ex\}$ |
|   | $(PC = su \wedge \neg ho \wedge \neg sa \rightarrow wo) \vee (PC = su \wedge \neg ho \wedge sa \rightarrow wo)$ | 0.135 | $\{\}$, $\{ch\}$, $\{su\}$, $\{ch, su\}$, $\{su, ex\}$, $\{ch, su, ex\}$ |
|   | $(PC = su \wedge ho \wedge \neg sa \rightarrow wo) \vee (PC = su \wedge ho \wedge sa \rightarrow wo)$ | 0.015 | $\{\}$, $\{ch\}$, $\{su\}$, $\{ho\}$, $\{ch, su\}$, $\{ch, ho\}$, $\{su, ho\}$, $\{su, ex\}$, $\{ho, ex\}$, $\{ch, su, ho\}$, $\{ch, su, ex\}$, $\{ch, ho, ex\}$, $\{su, ho, ex\}$, $\{ch, su, ho, ex\}$ |

Table 4: Induced DAFs of our example PrAAF with a probabilistic clique from Figure 8. With arrows in the possible world column we denote attacks and not implication.

have a strong relation with PLP; (e) we take advantage of the existing work from the PLP community to provide an efficient system that implements PrAAFs; (f) define the Probabilistic Aurgument Normal Form for PrAAFs and prove the same properties as in the Probabilistic Attack Normal Form; and finally, introduce a new probabilistic element, the Probabilistic Cliques, that transparently introduce dependent probabilistic arguments in constellation semantics PrAAFs.

Our motivation with this paper is to provide a simpler but powerful definition for constellation PrAAFs; furthermore, we give a clear insight in the constellation semantics and its restrictions from the point of view of generating possible worlds. Finally, by introducing probabilistic cliques we have extended the knowledge representation power of PrAAFs to combine both independent and dependent probabilistic elements in the same graph. We want to point out that one can define more probabilistic elements in a PrAAF in order to describe even more complex relations, for example: a probabilistic two way attack, a probabilistic element depended on a variable, mutually exclusive attacks, etc. The combination of these formulations would allow more complex probabilistic distributions to be described. A different approach would be to directly assign a probabilistic distribution to the desired graphs like in [20]. We want to point out that constellation approaches describe a probabilistic distribution over an exponential number of graphs making cumbersome to simulate them by enumerating each and every one of those.

For future work, we want to investigate to what degree the constellation semantics can contain epistemic constructs and what relations are between the two approaches; another topic for future research direction is, how to combine probabilities with coalitions [47]; furthermore, it is interesting to investigate what probabilistic inference tasks could be defined for PrAAFs; finally, we are planning to examine how the properties of DAF can be used to simplifying probabilistic inference in PrAAFs.

## Acknowledgments

We want to thank the reviewers for their excellent and constructive comments. This research has been partially supported by: "Argumentation 360", "RACRA18" (funded by Ricerca di Base 2017/2018) and supported by the project that has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 739578 (RISE Call: H2020-WIDESPREAD-01-2016-2017-TeamingPhase2) and the Government of the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development.

## Bibliography

[1] P. M. Dung, An argumentation-theoretic foundations for logic programming, J. Log. Program. 22 (2) (1995) 151–171.

[2] A. Bondarenko, F. Toni, R. A. Kowalski, An assumption-based framework for non-monotonic reasoning, in: Logic Programming and Non-monotonic Reasoning, (LPNMR), 1993, pp. 171–189.

[3] N. Oren, T. J. Norman, Semantics for evidence-based argumentation, in: Computational Models of Argument, (COMMA), 2008, pp. 276–284.

[4] Y. Wu, M. Caminada, A labelling-based justification status of arguments, Studies in Logic 3 (4) (2010) 12–29.

[5] P. Baroni, F. Cerutti, M. Giacomin, G. Guida, Afra: Argumentation framework with recursive attacks, International Journal of Approximate Reasoning 52 (1) (2011) 19–37.

[6] G. Brewka, S. Ellmauthaler, H. Strass, J. P. Wallner, S. Woltran, Abstract dialectical frameworks. an overview, FLAP 4 (8) (2017) .

[7] L. Amgoud, C. Cayrol, M.-C. Lagasquie-Schiex, On the bipolarity in argumentation frameworks, in: NMR, 2004.

[8] S. Modgil, H. Prakken, A general account of argumentation with preferences, Artificial Intelligence 195 (2013) 361–397.

[9] T. J. M. Bench-Capon, Persuasion in practical argument using value-based argumentation frameworks, J. Log. Comput. 13 (3) (2003) 429–448.

[10] S. Bistarelli, F. Rossi, F. Santini, A novel weighted defence and its relaxation in abstract argumentation, Int. J. Approx. Reasoning 92 (2018) 66–86.

[11] D. M. Gabbay, Fibring argumentation frames, Studia Logica 93 (2) (2009) 231.

[12] A. Hunter, A probabilistic approach to modelling uncertain logical arguments, International Journal of Approximate Reasoning 54 (1) (2013) 47–81.

[13] M. Thimm, A probabilistic semantics for abstract argumentation, in: European Conference on Artificial Intelligence, (ECAI), 2012, pp. 750–755.

[14] A. Hunter, M. Thimm, Probabilistic reasoning with abstract argumentation frameworks, J. Artif. Intell. Res. 59 (2017) 565–611.

[15] H. Li, N. Oren, T. J. Norman, Probabilistic argumentation frameworks, in: Theorie and Applications of Formal Argumentation, (TAFA), 2011, pp. 1–16.

[16] B. Fazzinga, S. Flesca, F. Parisi, On the complexity of probabilistic abstract argumentation, in: International Joint Conference on Artificial Intelligence, (IJCAI), 2013, pp. 898–904.

[17] D. Doder, S. Woltran, Probabilistic argumentation frameworks - A logical approach, in: Scalable Uncertainty Management, (SUM), 2014, pp. 134–147.

[18] P. Dondio, Toward a computational analysis of probabilistic argumentation frameworks, Cybernetics and Systems 45 (3) (2014) 254–278.

[19] B. Liao, K. Xu, H. Huang, Formulating semantics of probabilistic argumentation by characterizing subgraphs: Theory and empirical results, CoRR abs/1608.00302 (2016) .

[20] A. Hunter, M. Thimm, Optimization of dialectical outcomes in dialogical argumentation, International Journal of Approximate Reasoning 78 (2016) 73–102.

[21] A. Hunter, Some foundations for probabilistic abstract argumentation, in: Computational Models of Argument, (COMMA), 2012, pp. 117–128.

[22] M. W. A. Caminada, W. A. Carnielli, P. E. Dunne, Semi-stable semantics, Journal of Logic and Computation 22 (5) (2012) 1207–1254.

[23] S. A. Gaggl, S. Woltran, Cf2 semantics revisited, in: Computational Models of Argument, (COMMA), 2010, pp. 243–254.

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

[25] T. Mantadelis, G. Janssens, Dedicated tabling for a probabilistic setting, in: International Conference on Logic Programming, (ICLP), Vol. 7, 2010, pp. 124–133.

[26] F. Riguzzi, T. Swift, The PITA system: Tabling and answer subsumption for reasoning under uncertainty, CoRR abs/1107.4747 (2011) .

[27] A. Kimmig, B. Demoen, L. De Raedt, V. Santos Costa, R. Rocha, On the implementation of the probabilistic logic programming language ProbLog, Theory and Practice of Logic Programming (TPLP) 11 (2-3) (2011) 235–262.

[28] D. Fierens, G. Van Den Broeck, J. Renkens, D. Shterionov, B. Gutmann, I. Thon, G. Janssens, L. De Raedt, Inference and learning in probabilistic logic programs using weighted boolean formulas, Theory and Practice of Logic Programming (TPLP) 15 (3) (2014) 358–401.

[29] D. S. Shterionov, G. Janssens, Implementation and performance of probabilistic inference pipelines, in: Practical Aspects of Declarative Languages, (PADL), 2015, pp. 90–104.

[30] S. Bragaglia, F. Riguzzi, Approximate inference for logic programs with annotated disjunctions, in: Inductive Logic Programming, (ILP), 2011, pp. 30–37.

[31] J. Vlasselaer, G. Van den Broeck, A. Kimmig, W. Meert, L. De Raedt, Anytime inference in probabilistic logic programs with Tp-compilation, in: International Joint Conference on Artificial Intelligence, (IJCAI), 2015, pp. 1852–1858.

[32] T. Mantadelis, R. Rocha, Using iterative deepening for probabilistic logic inference, in: Practical Aspects of Declarative Languages, (PADL), 2017, pp. 198–213.

[33] E. Oikarinen, S. Woltran, Characterizing strong equivalence for argumentation frameworks, Artificial Intelligence 175 (14) (2011) 1985–2009.

[34] T. Sato, A statistical learning method for logic programs with distribution semantics, in: Proceedings of International Conference on Logic Programming: Proceedings of ICLP, 1995, pp. 715–729.

[35] L. De Raedt, A. Kimmig, Probabilistic (logic) programming concepts, Machine Learning 100 (1) (2015) 5–47.

[36] T. Mantadelis, G. Janssens, Nesting probabilistic inference, in: International Colloquium on Implementation of Constraint LOgic Programming Systems (CICLOPS), 2011.

[37] S. Bistarelli, F. Santini, Conarg: A constraint-based computational framework for argumentation systems, in: IEEE International Conference on Tools with Artificial Intelligence, (ICTAI), 2011, pp. 605–612.

[38] S. Bistarelli, T. Mantadelis, F. Santini, C. Taticchi, Probabilistic Argumentation Frameworks with MetaProbLog and ConArg, in: International Conference on Tools with Artificial Intelligence, (ICTAI), 2018, pp. 675–679.

[39] A. Kimmig, B. Gutmann, V. Santos Costa, Trading memory for answers: Towards tabling problog, in: International Workshop on Statistical Relational Learning, (SRL), 2009.

[40] T. Mantadelis, R. Rocha, Using iterative deepening for probabilistic logic inference, in: Practical Aspects of Declarative Languages, (PADL), 2017, pp. 198–213.

[41] N. J. Nilsson, Probabilistic logic, Artificial Intelligence 28 (1) (1986) 71–87.

[42] J. Vennekens, M. Denecker, M. Bruynooghe, CP-logic: A language of causal probabilistic events and its relation to logic programming, Theory and Practice of Logic Programming TPLP 9 (3) (2009) 245–308.

[43] D. Poole, The Independent Choice Logic and Beyond, 2008, pp. 222–243.

[44] B. Gutmann, On continuous distributions and parameter estimation in probabilistic logic programs (over continue verdelingen en het schatten van parameters in probabilistische logische programma's), Ph.D. thesis, Katholieke Universiteit Leuven (2011).

[45] L. R. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–286.

[46] B. Fazzinga, S. Flesca, F. Furfaro, Complexity of fundamental problems in probabilistic abstract argumentation: Beyond independence (extended abstract), in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 6362–6366.

[47] S. Bistarelli, F. Santini, Coalitions of arguments: An approach with constraint programming, Fundam. Inform. 124 (4) (2013) 383–401.