A Model-based Theorem Prover for Epistemic Graphs for Argumentation

Anthony Hunter[1] and Sylwia Polberg[2]

[1] Department of Computer Science, University College London, UK,
[2] School of Computer Science and Informatics, Cardiff University, UK.

Abstract. Epistemic graphs are a recent proposal for probabilistic argumentation that allows for modelling an agent's degree of belief in an argument and how belief in one argument may influence the belief in other arguments. These beliefs are represented by probability distributions and how they affect each other is represented by logical constraints on these distributions. Within the full language of epistemic constraints, we distinguish a restricted class which offers computational benefits while still being powerful enough to allow for handling of many other argumentation formalisms and that can be used in applications that, for instance, rely on Likert scales. In this paper, we propose a model-based theorem prover for reasoning with the restricted epistemic language.

Keywords: Probabilistic argumentation · Epistemic argumentation · Abstract argumentation.

1 Introduction

Both the constellations approach [14,9,11,5,16,4,6] and the epistemic approach [19,10,3,13,18,20] to probabilistic argumentation offer a valuable way to represent and reason with various aspects of uncertainty arising in argumentation. The epistemic uncertainty is seen as the degree to which an argument is believed or disbelieved, thus providing a more fine–grained alternative to the standard Dung's approaches when it comes to determining the status of a given argument. Following the results of an empirical study with participants [17], epistemic graphs have been introduced as a generalization of the epistemic approach to probabilistic argumentation [8,12].

In this approach, the graph is augmented with a set of epistemic constraints that can restrict the belief we have in an argument with a varying degree of specificity and state how beliefs in arguments influence each other. This is illustrated in Example 1. The graphs can therefore model both attack and support as well as relations that are neither positive nor negative. The flexibility of this approach allows us to both model the rationale behind the existing semantics as well as completely deviate from them when required. The fact that we can specify the rules under which arguments should be evaluated and that we can include constraints between unrelated arguments permits the framework to be more context–sensitive. It also allows for better modelling of imperfect agents, which can be important in multi–agent applications.

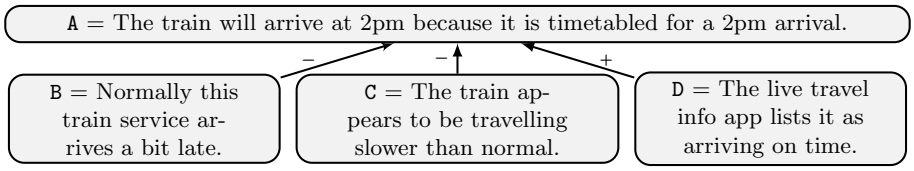

Fig. 1. Example of an epistemic graph. The + (resp. -) label denote support (resp. attack) relations. These are specified via the constraints given in Example 1.

Example 1. Consider the graph in Figure 1, and let us assume that if D is strongly believed, and B or C is strongly disbelieved, then A is strongly believed, whereas if D is believed, and B or C is disbelieved, then A is believed. Furthermore, if B or C is believed, then A is disbelieved. These constraints could be reflected by the following formulae: $\varphi_1 : p(\mathtt{D}) > 0.8 \wedge p(\mathtt{B} \vee \mathtt{C}) < 0.2 \Rightarrow p(\mathtt{A}) > 0.8$; $\varphi_2 : p(\mathtt{D}) > 0.5 \wedge p(\mathtt{B} \vee \mathtt{C}) \leq 0.5 \Rightarrow p(\mathtt{A}) > 0.5$; and $\varphi_3 : p(\mathtt{B} \wedge \mathtt{C}) > 0.5 \Rightarrow p(\mathtt{A}) < 0.5$.

Epistemic graphs are therefore a flexible and valuable tool for argumentation, and [12] has already provided methods for harnessing them in user modelling for persuasion dialogues. However, reasoning with the full epistemic language is non-trivial as the size of a probability distribution (i.e. the number of sets of arguments needing an assignment) is exponential w.r.t. the number of arguments, and there can potentially be infinitely many distributions satisfying a given set of constraints. As presented in [8], for certain applications a restricted form of logical constraint can be used, i.e. one where the probability values appearing in constraints and distributions come from a finite, restricted set of values. This may be appropriate if we want to represent beliefs in arguments as in a Likert scale [15],or we want to use epistemic graphs as a medium for existing extension-based or labeling-based methods. It also has the benefit of always producing a finite set of answers.

In order to reason with constraints based on a restricted set of values, we present a model-based theorem prover in this paper which can be used to check *(1)* whether constraints are consistent; *(2)* if one constraint entails another; and *(3)* find satisfying distributions. Our aim in this paper is to present a simple baseline system that can be implemented easily and used for small examples. This will help us understand some of the underlying issues in developing theorem provers for this formalism, and serve as a comparison for future systems.

We proceed as follows: Section 2 reviews epistemic graphs from [8]; Section 3 introduces a method for identifying the models for a constraint and Section 4 an algorithm for model-based reasoning (see [1] for proofs); and Section 5 discusses the contributions.

2 Epistemic Language

This section reviews the necessary basic definitions from [8]. We assume a directed graph $\mathcal{G} = (V, R)$, where each node in V denotes an argument (as illustrated by Figure 1), an edge in R denotes a relation between arguments and a

labeling $\mathcal{L} : R \to 2^{\{+,-,*\}} \setminus \{\varnothing\}$ tells us whether it is positive (+), negative (−), or neither (∗). We use $\mathsf{Nodes}(\mathcal{G})$ and $\mathsf{Arcs}(\mathcal{G})$ to denote V and R respectively. Epistemic graphs are simply labelled directed graphs equipped with a set of epistemic constraints (defined next) for capturing the influences between arguments. Both the labelled graph and the constraints provide information about the argumentation. In this paper, we focus on the constraints rather than on the full power of the graphs, and refer the readers to [8] for further details.

Like previously stated, the restricted epistemic language only allows values from a certain, finite set to appear in the formulae. However, in order for the approach to be coherent, this set should meet certain basic requirements. We thus use the notion of a (reasonable) restricted value set, which has to be closed under addition and subtraction (assuming the resulting value is still in the $[0, 1]$ interval) and contain value 1.

Definition 1. *A finite set of rational numbers from the unit interval Π is a* **reasonable restricted value set** *iff $1 \in \Pi$ and for any $x, y \in \Pi$ it holds that if $x + y \leq 1$, then $x + y \in \Pi$, and if $x - y \geq 0$, then $x - y \in \Pi$.*

We can also create subsets of this set according to a given inequality and threshold value as well as sequences of values that can be seen as satisfying a given arithmetical formula, which will become useful in the next sections:

Definition 2. *With $\Pi^x_\# = \{y \in \Pi \mid y \# x\}$ we denote the subset of Π obtained according to the value x and relationship $\# \in \{=, \neq, \geq, \leq, >, <\}$. The* **combination set** *for Π and a sequence of arithmetic operations $(*_1, \ldots, *_k)$ where $*_i \in \{+, -\}$ and $k \geq 0$ is defined as:*

$$\Pi^{x,(*_1,\ldots,*_k)}_\# = \begin{cases} \{(v) \mid v \in \Pi^x_\#\} & k = 0 \\ \{(v_1, \ldots, v_{k+1}) \mid v_i \in \Pi,\ v_1 *_1 \ldots *_k v_{k+1} \# x\} & \textit{otherwise} \end{cases}$$

Example 2. Let $\Pi_1 = \{0, 0.5, 0.75, 1\}$. We can observe that it is not a restricted value set, since $0.75 - 0.5 = 0.25$ is missing from Π_1. Its modification, $\Pi_2 = \{0, 0.25, 0.5, 0.75, 1\}$, is a restricted value set. The subsets of Π_2 for $x = 0.25$ under various inequalities are as follows: $\Pi_{2^>}^x = \{0.5, 0.75, 1\}$, $\Pi_{2^<}^x = \{0\}$, $\Pi_{2^\geq}^x = \{0.25, 0.5, 0.75, 1\}$, $\Pi_{2^\leq}^x = \{0, 0.25\}$, $\Pi_{2^\neq}^x = \{0, 0.5, 0.75, 1\}$, and $\Pi_{2^=}^x = \{0.25\}$.

Assume we have a reasonable restricted value set $\Pi_3 = \{0, 0.5, 1\}$, a sequence of operations $(+, -)$, an operator $=$ and a value $x = 1$. In order to find an appropriate combination set, we are simply looking for triples of values (τ_1, τ_2, τ_3) s.t. $x + y - z = 1$. This produces six possible value sequences, i.e. $\Pi_{2^=}^{1,(+,-)} = \{(0, 1, 0), (0.5, 0.5, 0), (0.5, 1, 0.5), (1, 0, 0), (1, 0.5, 0.5), (1, 1, 1)\}$.

2.1 Syntax and Semantics

Based on a given graph and restricted value set, we can now define the epistemic language. An epistemic formula can be seen as a propositional formula built out of components stating how the sums and/or subtractions of probabilities of argument terms should compare to values from Π.

Definition 3. *The **restricted epistemic language** based on $\mathcal{G}$ and a reasonable restricted value set Π is defined as follows:*

- *a* **term** *is a Boolean combination of arguments. We use $\vee$, $\wedge$ and $\neg$ as connectives and can derive secondary connectives, such as $\rightarrow$, as usual.* $\mathsf{Terms}(\mathcal{G})$ *denotes all the terms that can be formed from the arguments in $\mathcal{G}$.*
- *an* **operational formula** *is of the form $p(\alpha_i) *_1 \ldots *_{k-1} p(\alpha_k)$ where all $\alpha_i \in \mathsf{Terms}(\mathcal{G})$ and $*_j \in \{+, -\}$.* $\mathsf{OForm}(\mathcal{G})$ *denotes the set of all possible operational formulae of $\mathcal{G}$ and we read $p(\alpha)$ as the probability of α.*
- *an* **epistemic atom** *is of the form $\gamma \# x$ where $\# \in \{=, \neq, \geq, \leq, >, <\}$, $x \in \Pi$ and $\gamma \in \mathsf{OForm}(\mathcal{G})$.*
- *an* **epistemic formula** *is a Boolean combination of epistemic atoms.* $\mathsf{EForm}(\mathcal{G})$ *denotes the set of all possible epistemic formulae of $\mathcal{G}$.*

The full, unrestricted language simply permits x to be a rational value in the unit interval, hence we do not recall it here.

Example 3. Let $\Pi = \{0, 0.5, 1\}$. In the epistemic language restricted w.r.t. Π, we can only have atoms of the form $\beta \# 0$, $\beta \# 0.5$, and $\beta \# 1$, where $\beta \in \mathsf{OForm}(\mathcal{G})$ and $\# \in \{=, \neq, \geq, \leq, >, <\}$. From these atoms we compose epistemic formulae using the Boolean connectives, such as $p(\mathtt{A}) + p(\mathtt{B}) \leq 0.5 \wedge p(\mathtt{C}) = 0$.

The semantics for constraints come in the form of belief distributions, which assign probabilities to sets of arguments. Their restricted counterparts enforce that the assigned probabilities come from a restricted value set:

Definition 4. *A* **belief distribution** *on arguments is a function $P : 2^{\mathsf{Nodes}(\mathcal{G})} \rightarrow [0, 1]$ s.t. $\sum_{\Gamma \subseteq \mathsf{Nodes}(\mathcal{G})} P(\Gamma) = 1$. With $\mathsf{Dist}(\mathcal{G})$ we denote the set of all belief distributions on $\mathsf{Nodes}(\mathcal{G})$. P is* **restricted** *w.r.t.Π iff for every $X \subseteq \mathsf{Nodes}(\mathcal{G})$, $P(X) \in \Pi$[3]. With $\mathsf{Dist}(\mathcal{G}, \Pi)$ we denote the set of restricted distributions of $\mathcal{G}$.*

From the probability distribution, we can derive the probability of a term and therefore of an argument. Each $\Gamma \subseteq \mathsf{Nodes}(\mathcal{G})$ corresponds to an interpretation of arguments. We say that Γ *satisfies an argument* $\mathtt{A}$ and write $\Gamma \vDash \mathtt{A}$ iff $\mathtt{A} \in \Gamma$. Essentially $\vDash$ is a classical satisfaction relation and can be extended to complex terms as usual. For instance, $\Gamma \vDash \neg\alpha$ iff $\Gamma \nvDash \alpha$ and $\Gamma \vDash \alpha \wedge \beta$ iff $\Gamma \vDash \alpha$ and $\Gamma \vDash \beta$. With this, we can define the following:

Definition 5. *The* **probability of a term** *is defined as the sum of the probabilities (beliefs) of its models: $P(\alpha) = \sum_{\Gamma \subseteq \mathsf{Nodes}(\mathcal{G}) \text{ s.t. } \Gamma \vDash \alpha} P(\Gamma)$.*

We say that an agent believes a term α to some degree if $P(\alpha) > 0.5$, disbelieves α to some degree if $P(\alpha) < 0.5$, and neither believes nor disbelieves α when $P(\alpha) = 0.5$. Please observe that in this notation, $P(\mathtt{A})$ stands for the probability of a simple term $\mathtt{A}$ (i.e. sum of probabilities of all sets containing $\mathtt{A}$), which is different from $P(\{\mathtt{A}\})$, i.e. the probability assigned to set $\{\mathtt{A}\}$.

Using this, we can finally produce (restricted) satisfying distributions of a given atom, and therefore of a given formula:

[3] We note that this is a simpler, but still equivalent version of the notion in [8].

Definition 6. *Let* $\varphi : p(\alpha_i) \star_1 \ldots \star_{k-1} p(\alpha_k) \# b$ *be an epistemic atom. The* **satisfying distributions**, *or equivalently* **models**, *of* φ *are defined as* $\mathsf{Sat}(\varphi) = \{P' \in \mathsf{Dist}(\mathcal{G}) \mid P(\alpha_i) \star_1 \ldots \star_{k-1} P(\alpha_k) \# b\}$. *The* **restricted satisfying distribution** *of* φ *w.r.t.* Π *are defined as* $\mathsf{Sat}(\psi, \Pi) = \mathsf{Sat}(\psi) \cap \mathsf{Dist}(\mathcal{G}, \Pi)$.

The set of satisfying distributions for a given epistemic formula is as follows where ϕ and ψ are epistemic formulae: $\mathsf{Sat}(\phi \wedge \psi) = \mathsf{Sat}(\phi) \cap \mathsf{Sat}(\psi)$; $\mathsf{Sat}(\phi \vee \psi) = \mathsf{Sat}(\phi) \cup \mathsf{Sat}(\psi)$; and $\mathsf{Sat}(\neg\phi) = \mathsf{Sat}(\top) \setminus \mathsf{Sat}(\phi)$. For a set of epistemic formulae $\Phi = \{\phi_1, \ldots, \phi_n\}$, the set of satisfying distributions is $\mathsf{Sat}(\Phi) = \mathsf{Sat}(\phi_1) \cap \ldots \cap \mathsf{Sat}(\phi_n)$. The same holds for the restricted scenario.

Example 4. Let us assume we have a formula $\psi : p(\mathtt{A}) + p(\mathtt{B}) \leq 0.5$ on a graph s.t. $\{\mathtt{A}, \mathtt{B}\} = \mathsf{Nodes}(\mathcal{G})$. There can be infinitely many satisfying distributions of this formula, including P_1 s.t. $P_1(\varnothing) = 1$, P_2 s.t. $P_2(\varnothing) = P_2(\{\mathtt{A}\}) = 0.5$, P_3 s.t. $P_3(\varnothing) = P_3(\{\mathtt{B}\}) = 0.5$, or P_4 s.t. $P_4(\varnothing) = 0.68$, $P_4(\{\mathtt{A}\}) = 0.13$ and $P_4(\{\mathtt{B}\}) = 0.19$ (omitted sets are assigned 0). In contrast, a probability distribution P_5 s.t. $P_5(\{\mathtt{A}, \mathtt{B}\}) = 0.3$ and $P_5(\varnothing) = 0.7$ would not be satisfying. If we considered a restricted value set $\Pi = \{0, 0.5, 1\}$, then we could observe that P_1 to P_3 would be the all and only restricted satisfying distributions of ψ.

2.2 Epistemic Entailment Relation

In order to reason with the restricted epistemic language, we can use the consequence or the entailment relation. Given the focus of this paper, we will now recall the latter. From now on, unless stated otherwise, we will assume that the argumentation framework we are dealing with is finite and nonempty (i.e. the set of arguments in the graph is finite and nonempty).

Definition 7. *Let* Π *be a reasonable restricted value set,* $\psi \in \mathsf{EForm}(\mathcal{G}, \Pi)$ *an epistemic formula and* $\{\phi_1, \ldots, \phi_n\} \subseteq \mathsf{EForm}(\mathcal{G}, \Pi)$ *a set of epistemic formulae. The* **restricted epistemic entailment relation** *w.r.t.* Π, *denoted* $\models_\Pi$, *is defined as follows.*

$$\{\phi_1, \ldots, \phi_n\} \models_\Pi \psi \text{ iff } \mathsf{Sat}(\{\phi_1, \ldots, \phi_n\}, \Pi) \subseteq \mathsf{Sat}(\psi, \Pi)$$

Example 5. Consider $\Pi = \{0, 0.25, 0.5, 0.75, 1\}$ and restricted epistemic formulae $p(\mathtt{A}) + p(\neg\mathtt{B}) \leq 1$ and $p(\mathtt{A}) + p(\neg\mathtt{B}) \leq 0.75$. It holds that

$$\{p(\mathtt{A}) + p(\neg\mathtt{B}) \leq 0.75\} \models_\Pi p(\mathtt{A}) + p(\neg\mathtt{B}) \leq 1$$

It is worth noting how changing the restricted valued set affects the entailment. We can observe that a less restricted entailment (i.e. one with Π permitting more values) implies a more restricted one, but not necessarily the other way around, as seen in Example 6.

Proposition 1. *(From [8]) Let* $\Pi_1 \subseteq \Pi_2$ *be reasonable restricted value sets. For a set of epistemic formulae* $\Phi \subseteq \mathsf{EForm}(\mathcal{G}, \Pi_1)$, *and an epistemic formula* $\psi \in \mathsf{EForm}(\mathcal{G})$, *if* $\Phi \models_{\Pi_2} \psi$, *then* $\Phi \models_{\Pi_1} \psi$.

Example 6. Consider two formulae $\varphi_1 : p(\mathtt{A}) \neq 0.5$ and $\varphi_2 : p(\mathtt{A}) = 0 \vee p(\mathtt{A}) = 1$ and a reasonable restricted set $\Pi = \{0, 0.5, 1\}$. We can observe that $\mathsf{Sat}(\varphi_1, \Pi) = \mathsf{Sat}(\varphi_2, \Pi)$ and therefore $\{\varphi_1\} \Vdash_\Pi \varphi_2$. However, if we had set such as $\Pi' = \{0, 0.25, 0.5, 0.75, 1\}$, we could then consider a probability distribution P s.t. $P(\mathtt{A}) = 0.75$ in order to show that $\mathsf{Sat}(\varphi_1) \not\subseteq \mathsf{Sat}(\varphi_2)$.

3 Model-based Reasoning

A simple route to theorem proving is to use the definition of entailment. This involves identifying the models of the formulae by decomposing them to find the models of their subformulae, and then composing these sets of models to identify the models of the orignal formulae. We first define decomposition rules to split the formulae (Definition 8). These rules are used to reduce an epistemic formula to epistemic atoms of the form $p(\alpha) = v$ (if possible), and then finally to a set of models that satisfy the epistemic atom. Once we have decomposed a formula, we use the model propagation function (Definition 10) to combine the models of the epistemic atoms into models of the original formula.

Definition 8. *The* **decomposition rules** *are as follows where for each rule, the condition is an epistemic formula, and where Π is a reasonable restricted value set and $\# \in \{=, \neq, \geq, \leq, >, <\}$.*

- *The* **propositional rules** *are as follows where $x|y$ denotes that x is the left child and y is the right child, and from left to right, they are the* **conjunction**, **disjunction**, **implication**, *and* **negation** *rules.*

$$\frac{\phi \wedge \psi}{\phi \mid \psi} \qquad \frac{\phi \vee \psi}{\phi \mid \psi} \qquad \frac{\phi \rightarrow \psi}{\neg\phi \mid \psi} \qquad \frac{\neg\phi}{\phi}$$

- *The* **operational rules** *are defined as follows, where either $n > 0$ or $\#$ is different from $=$.*

$$\frac{p(\alpha_1) *_1 \ldots *_n p(\alpha_{n+1}) \# x}{\bigvee_{(v_1,\ldots,v_{n+1}) \in \Pi_{\#}^{x,(*_1,\ldots,*_n)}} (p(\alpha_1) = v_1 \wedge \ldots \wedge p(\alpha_k) = v_{n+1})} \quad \text{if } \Pi_{\#}^{x} \neq \varnothing$$

$$\frac{p(\alpha_1) *_1 \ldots *_n p(\alpha_{n+1}) \# x}{\varnothing} \qquad \text{otherwise}$$

- *The* **term rule** *is defined as follows.*

$$\frac{p(\alpha) = v}{\{P \in \mathsf{Dist}(\mathcal{G}, \Pi) \mid (\sum_{X \subseteq \mathsf{Nodes}(\mathcal{G}) \ s.t. \ X \vDash \alpha} P(X)) = v\}}$$

The decomposition of an epistemic formula using the above decomposition rules can be represented by a decomposition tree which we define next. For this, we assume that for a node n in a tree T, $\mathsf{Children}(n)$ is the set of children of n.

Definition 9. *A* **decomposition tree** *for an epistemic formula* $\phi \in \mathsf{EForm}(\mathcal{G})$ *is a tree where (1) the root is labelled with* ϕ*; (2) each non-leaf node is labelled with an epistemic formula* $\psi \in \mathsf{EForm}(\mathcal{G})$*; (3) each non-leaf node is associated with a decomposition rule such that the epistemic formula labelling the node satisfies the condition for the decomposition rule, and the child (or children in the case of the proposition rules) are obtained by the application of the decomposition rule; and (4) each leaf is a (possibly empty) set of models.* $\mathsf{Rule}(n)$ *denotes the decomposition rule that was applied to a non-leaf node* n*.*

Each decomposition tree is exhaustive, i.e. no further decomposition rules can be applied without violating the conditions of it being a decomposition tree. A possible decomposition tree can be seen in Figure 2 paired with Table 1.

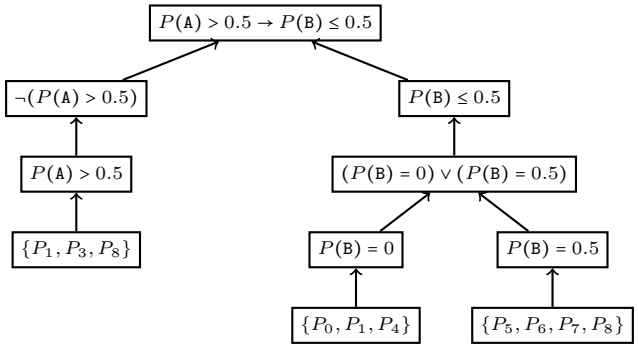

Fig. 2. A decomposition tree. Let P_0 to P_9 be defined as in Table 1. For the root, the set of models is $\{P_0, P_1, P_2, P_4, P_5, P_6, P_7, P_8, P_9\}$.

Table 1. Models for Figure 2 where $\Pi = \{0, 0.5, 1\}$ and $\mathsf{Nodes}(\mathcal{G}) = \{\mathtt{A}, \mathtt{B}\}$

	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
$\varnothing$	1	0	0	0	0.5	0.5	0.5	0	0	0
$\{\mathtt{A}\}$	0	1	0	0	0.5	0	0	0.5	0.5	0
$\{\mathtt{B}\}$	0	0	1	0	0	0.5	0	0.5	0	0.5
$\{\mathtt{A}, \mathtt{B}\}$	0	0	0	1	0	0	0.5	0	0.5	0.5

Each leaf of a decomposition tree is a, possibly empty, set of models (i.e. a set of probability distributions) that satisfy the epistemic formula at its parent node. In other words, the models of the leaf can be used to determine the models of its parent. Furthermore, for each non-leaf node, its models can be used to determine the models of its parent. So in general, for any non-leaf node, its models are a function of the models of its children, as we specify next.

Definition 10. *For a decomposition tree* T*, the* **model propagation function** *for* T*, denoted* Models*, is defined a follows,*

1. *If* $\mathsf{Rule}(n)$ *is the conjunction rule, and* $\mathsf{Children}(n) = \{n_1, n_2\}$*, then* $\mathsf{Models}(n) = \mathsf{Models}(n_1) \cap \mathsf{Models}(n_2)$.
2. *If* $\mathsf{Rule}(n)$ *is the disjunction or implication rule, and* $\mathsf{Children}(n) = \{n_1, n_2\}$*, then* $\mathsf{Models}(n) = \mathsf{Models}(n_1) \cup \mathsf{Models}(n_2)$.
3. *If* $\mathsf{Rule}(n)$ *is the negation rule, and* $\mathsf{Children}(n) = \{n_1\}$*, then* $\mathsf{Models}(n) = \mathsf{Sat}(\top, \Pi) \setminus \mathsf{Models}(n_1)$.
4. *If* $\mathsf{Rule}(n)$ *is the term rule or the operational rule, and* $\mathsf{Children}(n) = \{n_1\}$*, then* $\mathsf{Models}(n) = \mathsf{Models}(n_1)$.

For any given epistemic formula, the decomposition trees for the epistemic formula have the same set of leaves where $\mathsf{Leaves}(T)$ is the set of leaves in T.

Proposition 2. *If* T_1 *and* T_2 *are decomposition trees for* ϕ*, then* $\mathsf{Leaves}(T_1) = \mathsf{Leaves}(T_2)$.

Furthermore, the model propagation function ensures that the decompositions trees for an epistemic formula have the same set of models at the root.

Proposition 3. *If* T_1 *and* T_2 *are decomposition trees for* ϕ*, and the root of* T_1 *(respectively* T_2*) is* n_1 *(respectively* n_2*), then* $\mathsf{Models}(n_1) = \mathsf{Models}(n_2)$.

For each decomposition rule, the models of the epistemic formula in the condition of the rule are a function of the models in the consequent of the rule. For the conjunction (respectively disjunction) propositional decomposition rule, with condition ϕ, and consequent $\psi_1 \mid \psi_2$, $P \in \mathsf{Sat}(\phi, \Pi)$ iff $P \in \mathsf{Sat}(\psi_1, \Pi)$ and (respectively or) $P \in \mathsf{Sat}(\psi_2, \Pi)$. For the negation propositional decomposition rule, with condition ϕ, and consequent ψ, $P \in \mathsf{Sat}(\phi, \Pi)$ iff $P \notin \mathsf{Sat}(\psi, \Pi)$. For the term rule or the operational rule, the models of the condition of the rule are the models of the consequent. Hence, given a decomposition tree for an epistemic formula, the models of that formula are the models returned by backwards induction.

Proposition 4. *If* T *is a decomposition tree for epistemic formula* ϕ*, and the root of the tree is node* n*, then* $\mathsf{Sat}(\phi, \Pi) = \mathsf{Models}(n)$.

So constructing a decomposition tree is a method that is guaranteed to return exactly the models for the epistemic formula at the root.

4 Model-based Theorem Proving

Our proposal for model based theorem proving is based on the `Entailment` method given in Algorithm 1 which is defined in terms of the `GetModels`. The advantage of the algorithm is that it is straightforward to implement.

Proposition 5. *Algorithm 1 terminates.*

```
Entailment(φ, ψ)
        return GetModels(φ) ⊆ GetModels(ψ)

GetModels(φ);
    if φ = ψ1 ∧ ψ2 for some ψ1, ψ2
        return GetModels(ψ1) ∩ GetModels(ψ2)
    else if φ = ψ1 ∨ ψ2 for some ψ1, ψ2
        return GetModels(ψ1) ∪ GetModels(ψ2)
    else if φ = ψ1 → ψ2 for some ψ1, ψ2
        return GetModels(¬ψ1) ∪ GetModels(ψ2)
    else if φ = ¬ψ for some ψ
        return Dist(G, Π) ∖ GetModels(ψ)
    else if φ = p(α) = v for some α
        return {P ∈ Dist(G, Π) | (∑_{X⊆Nodes(G)} s.t. X⊨α P(X)) = v}
    else if φ = p(α1) ∗1 ... ∗n p(αn+1)#x for some α1, ..., αn+1
        if Π#^{x,(∗1,...,∗n)} = ∅
            return ∅
        else return ⋃_{(v1,...,vn+1)∈Π#^{x,(∗1,...,∗n)}} (⋂_{1≤i≤n+1} GetModels(p(αi) = vi))
```

Algorithm 1: Entailment which if the entailment holds, returns true, otherwise returns false.

However, the disadvantage of this algorithm is that it is computationally naive, and does not scale well, because it considers the potentially large number of probability distributions. In order to investigate the algorithm in practice, we implemented it in Python (see [2] for code), and ran an evaluation on a Windows 10 HP Pavilion Laptop (with AMD A10 2GHz processor and 8GB RAM) on a number of examples taken from [8]. For instance, for the following formulae, we obtained the results in Table 2 for time taken for entailment.

(1) $p(\mathtt{A} + \mathtt{B}) \leq 1$
(2) $p(\mathtt{A}) > 0.5 \rightarrow P(\mathtt{B}) \leq 0.5$
(3) $p(\mathtt{A}) < 0.9 \wedge p(\mathtt{A}) > 0.7$
(4) $p(\mathtt{A}) > 0.7 \wedge \neg(p(\mathtt{A}) \geq 0.9)$
(5) $(p(\mathtt{B}) < 0.5 \wedge p(\mathtt{C}) > 0.5 \wedge p(\mathtt{D}) > 0.5) \rightarrow p(\mathtt{B}) > 0.5$
(6) $(p(\mathtt{B}) < 0.5 \wedge p(\mathtt{C} \wedge \mathtt{D}) > 0.5) \rightarrow p(\mathtt{B}) > 0.5$
(7) $p(\mathtt{A}) \vee p(\mathtt{B}) \vee p(\mathtt{C}) \vee p(\mathtt{D}) > 0.5$
(8) $p(\mathtt{A} \vee \mathtt{B} \vee \mathtt{C} \vee \mathtt{D}) > 0.5$

Since the implementation is based on generating and manipulating sets of models, the number of models is the dominant factor in the running time. To illustrate this, we focus on the method in the implementation for generating the methods. For example, for generating the models for $|\Pi| = 5$, the running time with $|\mathsf{Nodes}(\mathcal{G})| = 2$ (respectively 3, 4, and 5) is 0.001 (respectively 0.032, 1.927, and 59.12) seconds, and so the theoretical results (that are discussed below) are reflected in the running time. Essentially, the implementation takes a brute-force approach since it generates all the models for the given set of arguments in the graph and the restricted value set, before decomposing the formulae and finding the models of the subformulae.

Table 2. Average running time in seconds for implementation of entailment on examples of formulae for each column where $\Pi = \{0, 0.25, 0.5, 0.75, 1\}$. Time is average of 10 runs for each pair. For each pair (x, y), x is the assumption and y is the conclusion of the entailment. For all pairs, entailment holds, except for $(6, 5)$ and $(8, 7)$.

	(1,2)	(2,1)	(3,4)	(4,3)	(5,6)	(6,5)	(7,8)	(8,7)
Time (secs)	0.037	0.036	0.008	0.010	1.023	1.032	137.6	122.0

For comparison, we look at the number of models that are generated in general. Given Π and $\mathsf{Nodes}(\mathcal{G})$, we can calculate the number of probability distributions for any language for epistemic formulae. For this, we say that a set of rational numbers Ξ is **compatible** with an integer n iff there is a bijection $f : \Xi \to \{0, 1, \ldots, n\}$ and a value $k \in \mathbb{N}$ such that for each $x \in \Xi$, $f(x) = kx$. For example, $\Xi = \{0, 0.5, 1\}$ is compatible with 2 and $\Xi = \{0, 0.25, 0.5, 0.75, 1\}$ is compatible with 4.

Lemma 1. *If Π is a reasonable restricted value set, then there is an integer n s.t. Π is compatible with n.*

Proposition 6. *Let Π be compatible with integer n. The cardinality of the set of probability distributions for Π and $\mathcal{G}$ is given by the following binomial coefficient (using the stars and bars method [7]) where $k = 2^{|\mathsf{Nodes}(\mathcal{G})|}$*

$$\binom{n+k-1}{n} = \frac{(n+k-1)!}{(k-1)!n!}$$

So, for a set $\Pi = \{0, 0.5, 1\}$ and $|\mathsf{Nodes}(\mathcal{G})| = 2$, we have $|\mathsf{Dist}(\mathcal{G}), \Pi| = 10$, for $\Pi = \{0, 0.25, 0.5, 0.75, 1\}$ and $|\mathsf{Nodes}(\mathcal{G})| = 2$, we have $|\mathsf{Dist}(\mathcal{G}, \Pi)| = 35$, and for $\Pi = \{0, 0.25, 0.5, 0.75, 1\}$ and $|\mathsf{Nodes}(\mathcal{G})| = 5$, we have $|\mathsf{Dist}(\mathcal{G}, \Pi)| = 52,360$.

5 Discussion

Epistemic graphs offer a rich formalism for modelling argumentation. There is some resemblance with variants of abstract argumentation such as ranking and weighted approaches, constrained argumentation frameworks, and weighted ADFs. However, the conceptual differences between epistemic probabilities and abstract weights lead to significant differences in modelling (see [8] for details). Also see [8] for a discussion of differences with Bayesian networks. In [8], a sound and complete proof theory is provided for constraints with restricted value sets but no algorithmic method is provided. In this paper, we have addressed this by giving a formal and transparent algorithmic method for reasoning with constraints. It is a practical alternative (for small examples) to the probabilistic optimization approach presented in [12]), and it can be used as a baseline system for which new algorithms can be compared. In future work, we will improve the efficiency of the algorithm (for example, by a lazy construction of models). We will also move beyond this baseline system by rewriting the constraints into a set of propositional clauses, and use a SAT solver.

References

1. http://www0.cs.ucl.ac.uk/staff/A.Hunter/papers/autoepigraphextra.pdf
2. http://www0.cs.ucl.ac.uk/staff/A.Hunter/papers/autoepigraph.py
3. Baroni, P., Giacomin, M., Vicig, P.: On rationality conditions for epistemic probabilities in abstract argumentation. In: Proceedings of COMMA'14. FAIA, vol. 266, pp. 121–132. IOS Press (2014)
4. Bistarelli, S., Mantadelis, T., Santini, F., Taticchi, C.: Probabilistic argumentation frameworks with metaproblog and conarg. In: Proceedings of ICTAI'18. pp. 675–679. IEEE (2018)
5. Doder, D., Woltran, S.: Probabilistic argumentation frameworks – a logical approach. In: Proceedings of SUM'14. LNCS, vol. 8720, pp. 134–147. Springer (2014)
6. Fazzinga, B., Flesca, S., Furfaro, F.: Complexity of fundamental problems in probabilistic abstract argumentation: Beyond independence. Artificial Intelligence **268**, 1–29 (2018)
7. Feller, W.: An Introduction to Probability Theory and Its Applications, vol. 1. Wiley, 2nd edn. (1950)
8. Hunter, A., Polberg, S., Thimm, M.: Epistemic graphs for representing and reasoning with positive and negative influences of arguments. ArXiv CoRR (2018), abs/1802.07489
9. Hunter, A.: Some foundations for probabilistic abstract argumentation. In: Proceedings of COMMA'12. FAIA, vol. 245, pp. 117–128. IOS Press (2012)
10. Hunter, A.: A probabilistic approach to modelling uncertain logical arguments. International Journal of Approximate Reasoning **54**(1), 47–81 (2013)
11. Hunter, A.: Probabilistic qualification of attack in abstract argumentation. International Journal of Approximate Reasoning **55**, 607–638 (2014)
12. Hunter, A., Polberg, S., Potyka, N.: Updating Belief in Arguments in Epistemic Graphs. In: Proceedings of KR'18. pp. 138–147. AAAI Press (2018)
13. Hunter, A., Thimm, M.: Probabilistic reasoning with abstract argumentation frameworks. Journal of Artificial Intelligence Research **59**, 565–611 (2017)
14. Li, H., Oren, N., Norman, T.J.: Probabilistic argumentation frameworks. In: Proceedings of TAFA'11. LNCS, vol. 7132, pp. 1–16. Springer (2011)
15. Likert, R.: A technique for the measurement of attitudes. Archives of Psychology **140**, 1–55 (1931)
16. Polberg, S., Doder, D.: Probabilistic abstract dialectical frameworks. In: Proceedings of JELIA'14. LNCS, vol. 8761, pp. 591–599. Springer (2014)
17. Polberg, S., Hunter, A.: Empirical evaluation of abstract argumentation: Supporting the need for bipolar and probabilistic approaches. International Journal of Approximate Reasoning **93**, 487 – 543 (2018)
18. Polberg, S., Hunter, A., Thimm, M.: Belief in attacks in epistemic probabilistic argumentation. In: Proceedings of SUM'17. LNCS, vol. 10564, pp. 223–236. Springer (2017)
19. Thimm, M.: A probabilistic semantics for abstract argumentation. In: Proceedings of ECAI'12. FAIA, vol. 242, pp. 750–755. IOS Press (2012)
20. Thimm, M., Polberg, S., Hunter, A.: Epistemic attack semantics. In: Proceedings of COMMA'18. FAIA, vol. 305, pp. 37–48. IOS Press (2018)