Ann Math Artif Intell
DOI 10.1007/s10472-015-9483-5

Belief revision in structured probabilistic argumentation

Model and application to cyber security

Paulo Shakarian[1] · Gerardo I. Simari[2] · Geoffrey Moores[3] · Damon Paulo[3] · Simon Parsons[4] · Marcelo A. Falappa[2] · Ashkan Aleali[1]

Abstract In real-world applications, knowledge bases consisting of all the available information for a specific domain, along with the current state of affairs, will typically contain contradictory data, coming from different sources, as well as data with varying degrees of uncertainty attached. An important aspect of the effort associated with maintaining such knowledge bases is deciding what information is no longer useful; pieces of information may be outdated; may come from sources that have recently been discovered to be of low

✉ Paulo Shakarian
shak@asu.edu

Gerardo I. Simari
gis@cs.uns.edu.ar

Geoffrey Moores
geoffrey.moores@usma.edu

Damon Paulo
damon.paulo@usma.edu

Simon Parsons
s.d.parsons@liverpool.ac.uk

Marcelo A. Falappa
mfalappa@cs.uns.edu.ar

Ashkan Aleali
aleali@asu.edu

[1] Arizona State University, Tempe, AZ, USA

[2] Department of Computer Science and Engineering, Universidad Nacional del Sur (UNS) and Institute for Computer Science and Engineering (CONICET-UNS), Bahía Blanca, Argentina

[3] Department of Electrical Engineering and Computer Science, U.S. Military Academy, West Point, NY, USA

[4] Department of Computer Science, University of Liverpool, Liverpool, UK

Published online: 25 September 2015

quality; or abundant evidence may be available that contradicts them. In this paper, we propose a probabilistic structured argumentation framework that arises from the extension of Presumptive Defeasible Logic Programming (PreDeLP) with probabilistic models, and argue that this formalism is capable of addressing these basic issues. The formalism is capable of handling contradictory and uncertain data, and we study non-prioritized belief revision over probabilistic PreDeLP programs that can help with knowledge-base maintenance. For belief revision, we propose a set of rationality postulates — based on well-known ones developed for classical knowledge bases — that characterize how these belief revision operations should behave, and study classes of operators along with theoretical relationships with the proposed postulates, including representation theorems stating the equivalence between classes of operators and their associated postulates. We then demonstrate how our framework can be used to address the attribution problem in cyber security/cyber warfare.

Keywords Argumentation · Belief revision · Probabilistic reasoning · Cyber security

Mathematics Subject Classification (2010) 68T30 · 68T27 · 68T37

1 Introduction

We begin by motivating our work, describing the most related work from the literature, introducing the cyber-attribution problem, and clarifying the contribution of the paper.

1.1 Motivation

In many real-world applications, knowledge bases consisting of all the information that is available about a specific domain, along with all the available information about the current state of affairs, will typically contain contradictory data. That is because the knowledge base will have been constructed using data from different sources that disagree. This data will also, typically, contain some measure of uncertainty. Thus, key problems that need to be addressed by formalisms for knowledge representation are the ability to handle contradictory information and to perform inference in the presence of uncertainty. In addition, in many cases it is necessary to update the knowledge in the knowledge base: for instance, pieces of information may be outdated, may come from sources which have recently been discovered to be of low quality, or there may be abundant evidence available that contradicts these pieces of information. In such cases, the knowledge base needs to be updated accordingly. A good example of how all of these requirements come together is provided by the scenario of determining the culprit of a *cyber attack*, an example that we will use in some detail to illustrate the ideas we develop in this paper. Here we provide a quick, motivating, sketch. The basic information in the scenario comes from a variety of different sources that only have a partial view of the domain, and since these sources disagree, having contradictory information in the knowledge base is unavoidable. In a cyber attack, it is not uncommon for the attacker to leave some false pieces of evidence with the goal of misleading the investigation, adding further contradictory information. None of the evidence that is gathered after an attack is conclusive, so there is uncertainty in the information that must be handled. Finally, since in responding to an attack new information is added to information that was gathered after previous attacks, it is necessary to update the knowledge base. In

particular, if new information contradicts old information, it may be necessary to perform belief revision to recover consistency of some parts of the knowledge-base.[1]

From this discussion we distill the requirements on any knowledge representation formalism that will be used in real-world applications. Such a formalism must be able to:

1. represent contradictory and uncertain information;
2. answer queries on a knowledge base; and
3. handle revisions to the knowledge base.

This paper presents a formalism called DeLP3E that meets all these requirements. A DeLP3E model consists of two parts, an *environmental model* (EM) and an *analytical model* (AM), which represent different aspects of a scenario. The idea is that the analytical model contains all the background information that is available for the analysis of the scenario. We envisage that this information is a combination of ontological information about the world, for example (to take the old example), "Tweety is a penguin", "penguins are birds" and "penguins do not fly", and commonsense information that is relevant, for example "birds generally fly". As can be seen from this small example, the AM can be inconsistent, and so we will choose a formal model for the AM that can cope with inconsistency. The environmental model is intended to contain evidence that has been collected about a specific situation (an instance of the more general model in the AM) about which queries will be answered. In the classic example, "Tweety is a penguin" would be an element of the EM, but the EM can also be more subtle than this, allowing for the representation of uncertain information. If we did not know for sure that Tweety was a penguin, but just had some suggestive evidence that this is so, we could, for example include in the EM the fact that "Tweety is a penguin" has a probability of 0.8 of being true. The EM is not limited to facts — we could also choose to model our evidence about Tweety with "Tweety is a bird" and "Tweety is black and white" and the rule that "Black and white birds have a probability of 0.8 of being penguins". A more complex pair of EM and AM, which relates to our motivating cyber security example, is given in Table 1.

The languages used in the AM and the EM are related through an *annotation function* (AF), which pairs formulae in the EM and the AM. Reasoning then consists of answering a query in the AM — when the AM is inconsistent, this will involve establishing the relevant consistent subset to answer the query computing the probability of the elements of the EM, and, through the annotation function, establishing the probabilities that correspond to the answer to the initial query. Thus, in the Tweety example, to answer a query about whether Tweety can fly, the AM would reason about this truth or falsity of the proposition "Tweety flies", the AF would identify which elements of the EM relate to this query, and the EM would provide a probability for these elements. The probability of the answer to the query, in this case either "Tweety flies" or "Tweety does not fly", could then be computed. The inference of this probability is what we call *entailment*.

In our vision, DeLP3E is less a specific formalism and more a *family* of formalisms where different formal models for handling uncertainty can be used for the EM, and different logical reasoning models can be used for the AM. In this paper, to make the discussion concrete, we make some specific choices. In particular, the EM is based on Nilsson's Probabilistic

[1] Below we discuss why we might want to carry out belief revision in a formalism that has the ability to handle some forms of inconsistency

Table 1 Examples of the kind of information that could be represented in the environmental and analytical models in a cyber-security application domain

Environmental model (EM)	Analytical model (AM)
Malware X was compiled on a system using the English language.	Malware X was compiled on a system in English-speaking country Y.
Country Y and country Z are currently at war.	Country Y has a motive to launch a cyber-attack against country Z
Malware W and malware X were created in a similar coding style.	Malware W and malware X are related.
Country Y has a significant investment in math-science-engineering (MSE) education.	Country Y has the capability to conduct a cyber-attack.

Logic [32], and the AM is based on the PreDeLP argumentation model from [30]. At the heart of PreDeLP is the notion of *presumptions*, elements of the knowledge base that can be presumed (assumed) to be true. This makes for a very natural connection to the EM, where the presumptions are the elements of the AM that connect (through the annotation function) to elements of the EM (as do the other elements of the AM). Thus, the presumptions will have a probability associated with them, and this is then used to establish the probability of the answer to the initial query.

This discussion has covered the requirement for DeLP3E to deal with inconsistency and uncertainty, and identified the need for inference. The final requirement is for the ability to revise the knowledge base, in particular the ability to perform belief revision in the sense of [1, 16, 17]. Given that belief revision is concerned with maintaining the consistency of a set of beliefs and that DeLP3E is built around an argumentation system that can handle inconsistency, at first glance it might not be obvious why belief revision will be required if these become inconsistent. However, on more reflection, it is clear that all three parts of a DeLP3E model — the environmental model, the analytical model, and the annotation function — may require revision, at least in the instantiation of DeLP3E that we consider here. The EM is underpinned by probability theory, and this places the constraint that the set of propositions used in the EM be consistent (a constraint that would not necessarily exist if we were to use a different uncertainty handling mechanism). The AM is built using PreDeLP, and though there can be inconsistency in some elements of a PreDeLP model, the strict rules and facts used to answer a specific query must be consistent, and so belief revision is required (if we built the AM using an argumentation system that only included defeasible knowledge, as in [36], belief revision would not be required). Finally, though there is never a strict requirement for belief revision of the AF, as we will discuss later, providing the ability to revise the annotation function can help us to avoid revising other aspects of the model, and this can be helpful.

1.2 Related work

The work that is closest to that reported here has been carried out in the intersection of belief revision and argumentation, and the work carried out in the combination of structured argumentation approaches with formalisms for probabilistic reasoning. We now discuss these

two points of contact with the existing literature; since, to the best of our knowledge, the combination we tackle in our work is completely novel, it is important to note that this discussion is necessarily short.

Belief revision studies changes to knowledge bases as a response to *epistemic inputs*. Traditionally, such knowledge bases can be either belief sets (sets of formulas closed under consequence) [1, 16, 17] or belief bases [20, 21] (which are not closed); since our end goal is to apply the results we obtain to real-world domains, here we focus on belief bases. In particular, our knowledge bases consist of logical formulas over which we apply argumentation-based reasoning and to which we couple a probabilistic model. The connection between belief revision and argumentation was first studied in [6]; see [12] and the further developments in [9]. Since then, the work that is most closely related to our approach is the development of the explanation-based operators of [11]. The main difference between that line of work and the one developed here arises from the interaction in our model between classical and probabilistic formalisms; to the best of our knowledge, this has not been tackled in the literature on combining argumentation and belief revision.

The study of argumentation systems together with probabilistic reasoning has recently received a lot attention, though a significant part of this recent work has concentrated on the combination of probability and abstract argumentation [14, 23, 28, 46]. There have, however, been several approaches that combine structured argumentation with models for reasoning under uncertainty; the first such approach to be proposed was [19][2] and several others followed, such as the possibilistic approach of [4], and the probabilistic logic-based approach of [24]. Similar to the difference between our approach and others on argumentation and belief revision, the main difference between these works and our own is that here we separate knowledge into the environmental model and the analytical model, where one part captures the probabilistic knowledge, and the other part the models knowledge that is not inherently probabilistic. This allows for a clear separation of interests between the two kinds of models. This approach is based on the similar approach developed for ontological languages in the Semantic Web (see [18], and references therein). The basic differences with that work is that the non-probabilistic part of the knowledge base corresponds to a classical ontology that is not inconsistency-tolerant, and that belief revision has not (again, to the best of our knowledge) been investigated in that formalism or others of its kind.

1.3 Application to the cyber-attribution problem

Cyber-attribution — the problem of determining who was responsible for a given cyber-operation, be it an incident of attack, reconnaissance, or information theft [39] — is an important issue. The difficulty of this problem stems not only from the amount of effort required to find forensic clues, but also the ease with which an attacker can plant false clues to mislead security personnel. Further, while techniques such as forensics and reverse-engineering [2], source tracking [47], honeypots [44], and sinkholing [37] are commonly employed to find evidence that can lead to attribution, it is unclear how this evidence is to be combined and reasoned about. In some cases, such evidence is augmented with normal intelligence collection, such as human intelligence (HUMINT), signals intelligence

[2]Krause et al. [26], which pre-dates [19], dealt with combining structured argumentation with abstract uncertainty measures and did not explicitly handle probability.

(SIGINT) and other means — this adds additional complications to the task of attributing a given operation.

In essence, cyber-attribution is a highly-technical intelligence analysis problem where an analyst must consider a variety of sources, each with its associated level of confidence, to provide a decision maker (e.g., a system administrator or Chief Information Officer) with insight into who conducted a given operation. Indeed, while previous cyber-attribution approaches only consider a single source of information, our approach takes into account multiple sources of information due to its ability to deal with inconsistency. As it is well-known that people's ability to conduct intelligence analysis is limited [22], and due to the highly technical nature of many cyber evidence-gathering techniques, an automated reasoning system would be best suited for the task. Such a system must be able to accomplish several goals:

- Reason about evidence in a formal, principled manner, i.e., relying on strong computational and mathematical foundations.
- Consider evidence for cyber attribution associated with some level of uncertainty (expressed via probabilities).
- Consider logical rules that allow for the system to draw conclusions based on certain pieces of evidence and iteratively apply such rules.
- Consider pieces of information that may not be compatible with each other, decide which information is most relevant, and express why.
- Attribute a given cyber-operation based on the above-described features and provide the analyst with the ability to understand how the system arrived at that conclusion.

The fit between these requirements and the abilities of DeLP3E led us to develop an extended example based around cyber-attribution[3] as a way of showcasing the functionality of DeLP3E. This example is given in Section 5.

1.4 Contribution of the paper

The main contribution of this paper is to present the DeLP3E framework, which combines structured argumentation and probability, and to discuss in detail how to perform belief revision in the context of this model. To our knowledge, this is the first paper to address the combination of structured probabilistic argumentation and belief revision. The paper brings together and extends the results of two papers that discussed structured probabilistic argumentation in respect to its application in cyber security — [40], which introduced the DeLP3E formalism (referred to there as P-PreDeLP) and annotation-function based belief revision, and [41], which studied a special case of the entailment query and showed how the framework can be applied to a cyber-attribution problem. Neither of these works include the more general entailment queries covered here in Section 3.3, the discussion of determining consistency from Section 4.1, or the AM-based belief revision introduced in Section 4.3. Further, this work includes complete proofs for all major theoretical results, as well as enhanced and expanded examples.

[3]The causality is a little more complicated than this sentence suggests. Indeed the cyber-attribution problem was the original motivation for the development of DeLP3E, and elements of the example evolved along with the formalism.

1.5 Structure of the paper

The structure of the paper broadly follows the three requirements identified above. First, in Section 2 we introduce the environmental and analytical model described above, where the environmental model makes use of Nilsson's probabilistic logic [32] and the analytical model builds upon PreDeLP [30]. The resulting framework is a general-purpose probabilistic argumentation language DeLP3E, which stands for Defeasible Logic Programming *with Presumptions and Probabilistic Environment*. This is formally laid out in Section 3. That section also studies the entailment problem for DeLP3E. Section 4 then provides the meat of the paper, discussing belief revision for the environmental model, the analytical model and the annotation function focusing on the study of non-prioritized belief revision operations. The paper suggests two sets of rationality postulates characterizing how such operations should behave, one for the analytical model and one for the annotation function (as we show, revising the environmental model is not sufficient to restore consistency). These postulates are based on the well-known approach proposed in [20] for non-prioritized belief revision in classical knowledge bases — and studies two classes of operators and their theoretical relationships with the proposed postulates, concluding with representation theorems for each class. Section 5 then walks through an extended example of the use of DeLP3E in the context of cyber-attribution. Section 6 concludes.

2 Preliminaries

This section presents the main componenets of the DeLP3E framework, the environmental model and the analytical model.

2.1 Basic language

We assume sets of variables and constants, denoted with $\mathbf{V}$ and $\mathbf{C}$, respectively. In the rest of this paper, we will follow the convention from the logic programming literature and use capital letters to represent variables (e.g., X, Y, Z) and lowercase letters to represent constants.

The next component of the language is a set of predicate symbols. Each predicate symbol has an arity bounded by a constant value; the EM and AM use separate sets of predicate symbols, denoted with $\mathbf{P}_{EM}$, $\mathbf{P}_{AM}$, respectively — the two models can, however, share variables and constants.

As usual, a *term* is composed of either a variable or a constant. Given terms $t_1, ..., t_n$ and n-ary predicate symbol p, $p(t_1, ..., t_n)$ is called an *atom*; if $t_1, ..., t_n$ are constants, then the atom is said to be *ground*. The sets of all ground atoms for the EM and AM are denoted with $\mathbf{G}_{EM}$ and $\mathbf{G}_{AM}$, respectively.

Given a set of ground atoms, a *world* is any subset of atoms — those that belong to the set are said to be *true* in the world, while those that do not are *false*. Therefore, there are $2^{|\mathbf{G}_{EM}|}$ possible worlds in the EM and $2^{|\mathbf{G}_{AM}|}$ worlds in the AM; these sets are denoted with $\mathcal{W}_{EM}$ and $\mathcal{W}_{AM}$, respectively. In order to avoid worlds that do not model possible situations given a particular domain, we include *integrity constraints* of the form $\mathsf{oneOf}(\mathcal{A}')$, where $\mathcal{A}\prime$ is a subset of ground atoms. Intuitively, such a constraint states that any world where more than one of the atoms from set $\mathcal{A}'$ appears is invalid. We use $\mathbf{IC}_{EM}$ and $\mathbf{IC}_{AM}$ to denote the sets of integrity constraints for the EM and AM, respectively, and the sets of worlds that conform to these constraints is denoted with $\mathcal{W}_{EM}(\mathbf{IC}_{EM})$ and $\mathcal{W}_{AM}(\mathbf{IC}_{AM})$, respectively.

Finally, logical formulas arise from the combination of atoms using the traditional connectives ($\wedge$, $\vee$, and $\neg$). As usual, we say that a world w *satisfies* formula f, written $w \models f$, iff: (i) If f is an atom, then $w \models f$ iff $f \in w$; (ii) if $f = \neg f'$ then $w \models f$ iff $w \not\models f'$; (iii) if $f = f' \wedge \theta''$ then $w \models f$ iff $w \models f'$ and $w \models \theta''$; and (iv) if $f = f' \vee \theta''$ then $w \models f$ iff $w \models f'$ or $w \models \theta''$. We use the notation $form_{EM}$, $form_{AM}$ to denote the set of all possible (ground) formulas in the EM and AM, respectively.

Example 1 Thus, the following are terms

$$\begin{array}{cccccccc} a & b & c & d & e & f & p(X) \\ g & h & i & j & k & & p(a) \end{array}$$

and the following are formulae using those terms:

$$\begin{array}{ccc} a & d \wedge e & k \\ b & f \wedge g \wedge h & \\ c & i \vee \neg j & \end{array}$$

2.2 Environmental model

The EM is used to describe the probabilistic knowledge that we have about the domain. In general, the EM contains knowledge such as evidence, uncertain facts, or knowledge about agents and systems. Here we base the EM on the probabilistic logic of [32], which we now briefly review.

Definition 1 Let f be a formula over $\mathbf{P}_{EM}$, $\mathbf{V}$, and $\mathbf{C}$, $p \in [0, 1]$, and $\epsilon \in [0, \min(p, 1-p)]$. A *probabilistic formula* is of the form $f : p \pm \epsilon$. A set $\mathcal{K}_{EM}$ of probabilistic formulas is called a *probabilistic knowledge base*.

In the above definition, the number ϵ is referred to as an *error tolerance*. Intuitively, the probabilistic formula $f : p \pm \epsilon$ is interpreted as "formula f is true with probability between $p-\epsilon$ and $p+\epsilon$". Note that there are no further constraints over this interval apart from those imposed by other probabilistic formulas in the knowledge base. The uncertainty regarding the probability values stems from the fact that certain assumptions (such as probabilistic independence between all formulae) may not hold in the environment being modeled.

Example 2 Consider the following set $\mathcal{K}_{EM}$:

$$\begin{array}{llll} f_1 = a : 0.8 \pm 0.1 & f_4 = d \wedge e : 0.7 \pm 0.2 & f_7 = k : 1 \pm 0 \\ f_2 = b : 0.2 \pm 0.1 & f_5 = f \wedge g \wedge h : 0.6 \pm 0.1 & f_8 = a \wedge b : 0.4 \pm 0.1 \\ f_3 = c : 0.8 \pm 0.1 & f_6 = i \vee \neg j : 0.9 \pm 0.1 & \end{array}$$

Throughout the paper, we also use $\mathcal{K}'_{EM} = \{f_1, f_2, f_3\}$

A set of probabilistic formulas describes a set of possible probability distributions Pr over the set $\mathcal{W}_{EM}(\mathbf{IC}_{EM})$. We say that probability distribution Pr *satisfies* probabilistic formula $f : p \pm \epsilon$ iff:

$$p - \epsilon \leq \sum_{w \in \mathcal{W}_{EM}(\mathbf{IC}_{EM}), w \models f} Pr(w) \leq p + \epsilon.$$

A probability distribution over $\mathcal{W}_{EM}(\mathbf{IC}_{EM})$ *satisfies* $\mathcal{K}_{EM}$ iff it satisfies all probabilistic formulas in $\mathcal{K}_{EM}$.

Given a probabilistic knowledge base and a (non-probabilistic) formula q, the *maximum entailment* problem seeks to identify real numbers p, ϵ such that all valid probability distributions Pr that satisfy $\mathcal{K}_{EM}$ also satisfy $q : p \pm \epsilon$, and there does not exist p', ϵ' s.t. $[p - \epsilon, p + \epsilon] \supset [p' - \epsilon', p' + \epsilon']$, where all probability distributions Pr that satisfy $\mathcal{K}_{EM}$ also satisfy $q : p' \pm \epsilon'$. In order to solve this problem we must solve the linear program defined below.

Definition 2 Given a knowledge base $\mathcal{K}_{EM}$ and a formula q, we have a variable x_i for each $w_i \in \mathcal{W}_{EM}(\mathbf{IC}_{EM})$. Each variable x_i corresponds with the probability of w_i occurring.

- For each $f_j : p_j \pm \epsilon_j \in \mathcal{K}_{EM}$, there is a constraint of the form: $p_j - \epsilon_j \leq \sum_{w_i \in \mathcal{W}_{EM}(\mathbf{IC}_{EM})\ s.t.\ w_i \models f_j} x_i \leq p_j + \epsilon_j$.
- We also have the constraint: $\sum_{w_i \in \mathcal{W}_{EM}(\mathbf{IC}_{EM})} x_i = 1$.
- The objective is to minimize the function: $\sum_{w_i \in \mathcal{W}_{EM}(\mathbf{IC}_{EM})\ s.t.\ w_i \models q} x_i$.

We use the notation EP-LP-MIN$(\mathcal{K}_{EM}, q)$ to refer to the value of the objective function in the solution to the EM-LP-MIN constraints.

The next step is to solve the linear program a second time, but this time maximizing the objective function (we shall refer to this as EM-LP-MAX) — let ℓ and u be the results of these operations, respectively. In [32], it is shown that:

$$\epsilon = \frac{u - \ell}{2} \text{ and } p = \ell + \epsilon$$

is the solution to the maximum entailment problem. We note that although the above linear program has an exponential number of variables in the worst case (i.e., no integrity constraints), the presence of constraints has the potential to greatly reduce this space. Further, there are also good heuristics (cf. [25, 42]) that have been shown to provide highly accurate approximations with a reduced-size linear program.

Example 3 Consider KB $\mathcal{K}'_{EM}$ from Example 2 and a set of ground atoms restricted to those that appear in that program; we have the following worlds:

$$\begin{array}{llll} w_1 = \{a, b, c\} & w_2 = \{a, b\} & w_3 = \{a, c\} & w_4 = \{b, c\} \\ w_5 = \{b\} & w_6 = \{a\} & w_7 = \{c\} & w_8 = \emptyset \end{array}$$

and suppose we wish to compute the probability for formula $q = a \vee c$.

For each formula in $\mathcal{K}_{EM}$ we have a constraint, and for each world above we have a variable. An objective function is created based on the worlds that satisfy the query formula (in this case, worlds $w_1, w_2, w_3, w_4, w_6, w_7$). Solving EP-LP-MAX$(\mathcal{K}'_{EM}, q)$ and EP-LP-MIN$(\mathcal{K}'_{EM}, q)$, we obtain the solution 0.9 ± 0.1. Hence, EP-LP-MAX$(\mathcal{K}'_{EM}, q)$ can be written as follows:

$$\begin{array}{rcl} \max & x_1 + x_2 + x_3 + x_4 + x_6 + x_7 & w.r.t. : \\ 0.7 \leq & x_1 + x_2 + x_3 + x_6 & \leq 0.9 \\ 0.1 \leq & x_1 + x_2 + x_4 + x_5 & \leq 0.3 \\ 0.8 \leq & x_1 + x_3 + x_4 + x_7 & \leq 1 \\ & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 & = 1 \end{array}$$

From this, we can solve EP-LP-MAX$(\mathcal{K}'_{EM}, q)$ and, after an easy modification, EP-LP-MIN $(\mathcal{K}'_{EM}, q)$, and obtain the solution 0.9 ± 0.1.

2.3 Analytical model

The analytical model contains information that a user may conclude based on the information in the environmental model. While the EM contains information that can have probabilities associated with it, statements in the AM can be either true or false depending on a certain combination (or several possible combinations) of statements from the EM.

For the AM, we choose to represent information using a structured argumentation framework [34] since this kind of formalism meets the representational requirements discussed in the introduction. Unlike the EM, which describes probabilistic information about the state of the real world, the AM must allow for competing ideas. Therefore, it must be able to represent contradictory information. The algorithmic approach we shall later describe allows for the creation of *arguments* based on the AM that may "compete" with each other to answer a given query. In this competition — known as a *dialectical process* — one argument may defeat another based on a *comparison criterion* that determines the prevailing argument. Resulting from this process, certain arguments are *warranted* (those that are not *defeated* by other arguments), thereby providing a suitable explanation for the answer to a given query.

The transparency provided by the system can allow knowledge engineers and users of the system to identify potentially incorrect input information and fine-tune the models or, alternatively, collect more information. In short, argumentation-based reasoning has been studied as a natural way to manage a set of inconsistent information — it is the way humans settle disputes. As we will see, another desirable characteristic of (structured) argumentation frameworks is that, once a conclusion is reached, we are left with an explanation of *how we arrived at it* and information about why a given argument is warranted; this is very important information for users to have.

The formal model that we use for the AM is Defeasible Logic Programming with Presumptions (PreDeLP) [30], a formalism combining logic programming with defeasible argumentation. Here, we briefly recall the basics of PreDeLP— we refer the reader to [15, 30] for the complete presentation. Formally, we use the notation

$$\Pi_{AM} = (\Theta, \Omega, \Phi, \Delta)$$

to denote a PreDeLP program, where Ω is a set of strict rules, Θ is a set of facts, Δ is a set of defeasible rules, and Φ is a set of presumptions. We now define these constructs formally.

Facts (Θ) are ground literals representing atomic information or its negation, using strong negation "¬". Note that all of the literals in our framework must be formed with a predicate from the set $\mathbf{P}_{AM}$. Note that information in the form of facts cannot be contradicted. We will use the notation $[\Theta]$ to denote the set of all possible facts.

Strict rules (Ω) represent non-defeasible cause-and-effect information that resembles an implication (though the semantics is different since the contrapositive does not hold) and are of the form $L_0 \leftarrow L_1, \ldots, L_n$, where L_0 is a ground literal and $\{L_i\}_{i>0}$ is a set of ground literals. We will use the notation $[\Omega]$ to denote the set of all possible strict rules.

Presumptions (Φ) are ground literals of the same form as facts, except that they are not taken as being true but rather are defeasible, which means that they can be contradicted. Presumptions are denoted in the same manner as facts, except that the symbol $\prec$ is added. We note that the presumptions cannot be treated as special cases of the defeasible rules. The

intuition of the presumption is that outside of other criteria, arguments with more presumptions should be defeated by arguments with less presumption which is not necessarily the case if the presumption is expressed as a defeasible rule. As shown in [30] the treatment of presumptions in this manner also necessitates an extension to generalized specificity. See [30] for further details.

Defeasible rules (Δ) represent tentative knowledge that can be used if nothing can be posed against it. Just as presumptions are the defeasible counterpart of facts, defeasible rules are the defeasible counterpart of strict rules. They are of the form $L_0 \prec L_1, \ldots, L_n$, where L_0 is a ground literal and $\{L_i\}_{i>0}$ is a set of ground literals. In both strict and defeasible rules, *strong negation* is allowed in the head of rules, and hence may be used to represent contradictory knowledge.

Even though the above constructs are ground, we allow for schematic versions with variables that are used to represent sets of ground rules. In Fig. 1, we provide an example Π_{AM} of a ground knowledge base. (Fig. 7 on Page 42 gives an example of a non-ground knowledge base.)

Arguments Given a query in the form of a ground atom, the goal is to derive arguments for and against its validity — derivation follows the mechanism of logic programming [29]. Since rule heads can contain strong negation, it is possible to defeasibly derive contradictory literals from a program. For the treatment of contradictory knowledge, PreDeLP incorporates a defeasible argumentation formalism that allows the identification of the pieces of knowledge that are in conflict and, through the previously mentioned dialectical process, decides which information prevails as warranted. This dialectical process involves the construction and evaluation of arguments, building a *dialectical tree* in the process. Arguments are formally defined next.

Definition 3 An *argument* $\langle \mathcal{A}, L \rangle$ for a literal L is a pair of the literal and a (possibly empty) set of the AM ($\mathcal{A} \subseteq \Pi_{AM}$) that provides a minimal proof for L meeting the following requirements: (i) L is defeasibly derived from $\mathcal{A}$; (ii) $\Omega \cup \Theta \cup \mathcal{A}$ is not contradictory; and (iii) $\mathcal{A}$ is a minimal subset of $\Delta \cup \Phi$.

Literal L is called the *conclusion* supported by the argument, and $\mathcal{A}$ is the *support* of the argument. An argument $\langle \mathcal{B}, L \rangle$ is a *subargument* of $\langle \mathcal{A}, L' \rangle$ iff $\mathcal{B} \subseteq \mathcal{A}$. An argument $\langle \mathcal{A}, L \rangle$ is *presumptive* iff $\mathcal{A} \cap \Phi$ is not empty. We will also use $\Omega(\mathcal{A}) = \mathcal{A} \cap \Omega$, $\Theta(\mathcal{A}) = \mathcal{A} \cap \Theta$,

Θ :	$\theta_{1a} =$	p	$\theta_{1b} =$	q	$\theta_2 =$	r
Ω :	$\omega_{1a} =$	$\neg s \leftarrow t$	$\omega_{2a} =$	$s \leftarrow p, u, r, v$		
	$\omega_{1b} =$	$\neg t \leftarrow s$	$\omega_{2b} =$	$t \leftarrow q, w, x, v$		
Φ :	$\phi_1 =$	$y \prec$	$\phi_2 =$	$v \prec$	$\phi_3 =$	$\neg z \prec$
Δ :	$\delta_{1a} =$	$s \prec p$	$\delta_2 =$	$s \prec u$	$\delta_{5a} =$	$\neg u \prec \neg z$
	$\delta_{1b} =$	$t \prec q$	$\delta_3 =$	$s \prec r, v$	$\delta_{5b} =$	$\neg w \prec \neg n$
			$\delta_4 =$	$u \prec y$		

Fig. 1 A ground argumentation framework

$\Delta(\mathcal{A}) = \mathcal{A} \cap \Delta$, and $\Phi(\mathcal{A}) = \mathcal{A} \cap \Phi$. For convenience, we may sometimes call an argument by its support. (e.g. argument $\mathcal{A}$ instead of argument $\langle \mathcal{A}, L \rangle$.

Our definition differs slightly from that of [43], where DeLP is introduced, as we include strict rules and facts as part of arguments — this is due to the fact that in our framework, the components of an argument can only be used in certain environmental conditions. Hence, a fact may be true in one EM world and not another, and so different sets of strict rules and facts may be applicable to different arguments. This is in contrast to DeLP when the same set of strict rules and facts can be applied to any argument and so do not have to be explicitly listed.. We discuss this further in Section 3 (page 16).

Definition 4 A literal is derived from an argument if it appears as a fact or a presumption in the argument or appears in the head of a strict rule or a defeasible rule where all the literals in the body of that strict rule or defeasible rule are derived from that argument.

Example 4 Figure 2 shows example arguments based on the knowledge base from Fig. 1. Note that $\langle \mathcal{A}_5, u \rangle$ is a sub-argument of $\langle \mathcal{A}_2, s \rangle$ and $\langle \mathcal{A}_3, s \rangle$.

Given an argument $\langle \mathcal{A}_1, L_1 \rangle$, counter-arguments are arguments that contradict it. Argument $\langle \mathcal{A}_2, L_2 \rangle$ is said to *counterargue* or *attack* $\langle \mathcal{A}_1, L_1 \rangle$ at a literal L' iff there exists a subargument $\langle \mathcal{A}, L'' \rangle$ of $\langle \mathcal{A}_1, L_1 \rangle$ such that the set $\Omega(\mathcal{A}_1) \cup \Omega(\mathcal{A}_2) \cup \Theta(\mathcal{A}_1) \cup \Theta(\mathcal{A}_2)$ is inconsistent.

Example 5 Consider the arguments from Example 4. The following are some of the attack relationships between them: $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$, and $\mathcal{A}_4$ all attack $\mathcal{A}_6$; $\mathcal{A}_5$ attacks $\mathcal{A}_7$; and $\mathcal{A}_7$ attacks $\mathcal{A}_2$.

A *proper defeater* of an argument $\langle A, L \rangle$ is a counter-argument that — by some criterion — is considered to be better than $\langle A, L \rangle$; if the two are incomparable according to this criterion, the counterargument is said to be a *blocking* defeater. An important characteristic of PreDeLP is that the argument comparison criterion is modular, and thus the most appropriate criterion for the domain that is being represented can be selected; the default criterion used in classical defeasible logic programming (from which PreDeLP is derived) is *generalized specificity* [45], though an extension of this criterion is required for arguments using presumptions [30]. We briefly recall this criterion next — the first definition is for generalized specificity, which is subsequently used in the definition of presumption-enabled specificity.

Definition 5 (Generalized Specificity) Let $\Pi_{AM} = (\Theta, \Omega, \Phi, \Delta)$ be a PreDeLP program and let $\mathcal{F}$ be the set of all literals that have a defeasible derivation from Π_{AM}. An

$\langle \mathcal{A}_1, s \rangle$	$\mathcal{A}_1 = \{\theta_{1a}, \delta_{1a}\}$	$\langle \mathcal{A}_2, s \rangle$	$\mathcal{A}_2 = \{\phi_1, \phi_2, \delta_4, \omega_{2a}, \theta_{1a}, \theta_2\}$
$\langle \mathcal{A}_3, s \rangle$	$\mathcal{A}_3 = \{\phi_1, \delta_2, \delta_4\}$	$\langle \mathcal{A}_4, s \rangle$	$\mathcal{A}_4 = \{\phi_2, \delta_3, \theta_2\}$
$\langle \mathcal{A}_5, u \rangle$	$\mathcal{A}_5 = \{\phi_1, \delta_4\}$	$\langle \mathcal{A}_6, \neg s \rangle$	$\mathcal{A}_6 = \{\delta_{1b}, \theta_{1b}, \omega_{1a}\}$
$\langle \mathcal{A}_7, \neg u \rangle$	$\mathcal{A}_7 = \{\phi_3, \delta_{5a}\}$		

Fig. 2 Example ground arguments from the framework of Fig. 1

argument $\langle \mathcal{A}_1, L_1 \rangle$ is *preferred to* $\langle \mathcal{A}_2, L_2 \rangle$, denoted with $\mathcal{A}_1 \succ_{PS} \mathcal{A}_2$ iff the two following conditions hold:

(1) For all $H \subseteq \mathcal{F}$, $\Omega \cup H$ is non-contradictory: if there is a derivation for L_1 from $\Omega \cup H \cup DR(\mathcal{A}_1)$, and there is no derivation for L_1 from $\Omega \cup H$, then there is a derivation for L_2 from $\Omega \cup H \cup DR(\mathcal{A}_2)$.
(2) There is at least one set $H' \subseteq \mathcal{F}$, $\Omega \cup H'$ is non-contradictory, such that there is a derivation for L_2 from $\Omega \cup H' \cup DR(\mathcal{A}_2)$, there is no derivation for L_2 from $\Omega \cup H'$, and there is no derivation for L_1 from $\Omega \cup H \cup DR(\mathcal{A}_1)$.

Intuitively, the principle of specificity says that, in the presence of two conflicting lines of argument about a proposition, the one that uses more of the available information is more convincing. Considering the Tweety example again; there are arguments stating both that Tweety flies (because it is a bird) and that Tweety doesn't fly (because it is a penguin). The latter argument uses more information about Tweety — it is more specific because it is information that Tweety is not just a bird, but is a penguin-bird, the subset of birds that are penguins — and is thus the stronger of the two.

Definition 6 (**Presumption-enabled Specificity** [30]) Given PreDeLP program $\Pi_{AM} = (\Theta, \Omega, \Phi, \Delta)$, an argument $\langle \mathcal{A}_1, L_1 \rangle$ is *preferred to* $\langle \mathcal{A}_2, L_2 \rangle$, denoted with $\mathcal{A}_1 \succ \mathcal{A}_2$ iff any of the following conditions hold:

(1) $\langle \mathcal{A}_1, L_1 \rangle$ and $\langle \mathcal{A}_2, L_2 \rangle$ are both factual, which is an argument using none of the presumptions or defeasible rules and $\langle \mathcal{A}_1, L_1 \rangle \succ_{PS} \langle \mathcal{A}_2, L_2 \rangle$.
(2) $\langle \mathcal{A}_1, L_1 \rangle$ is a factual argument and $\langle \mathcal{A}_2, L_2 \rangle$ is a presumptive argument, which is an argument using at least one of the presumptions or defeasible rules.
(3) $\langle \mathcal{A}_1, L_1 \rangle$ and $\langle \mathcal{A}_2, L_2 \rangle$ are presumptive arguments, and
    (a) $\Phi(\mathcal{A}_1) \subset \Phi(\mathcal{A}_2)$ or,
    (b) $\Phi(\mathcal{A}_1) = \Phi(\mathcal{A}_2)$ and $\langle \mathcal{A}_1, L_1 \rangle \succ_{PS} \langle \mathcal{A}_2, L_2 \rangle$.

Generally, if $\mathcal{A}$ and $\mathcal{B}$ are arguments with rules X and Y, respectively and $X \subset Y$, then $\mathcal{A}$ is stronger than $\mathcal{B}$. This also holds when $\mathcal{A}$ and $\mathcal{B}$ use presumptions P_1 and P_2, resp., and $P_1 \subset P_2$.

Example 6 The following are some relationships between arguments from Example 4, based on Definitions 5 and 6.

$\mathcal{A}_1$ and $\mathcal{A}_6$ are incomparable (blocking defeaters);
$\mathcal{A}_6 \succ \mathcal{A}_2$, and thus $\mathcal{A}_6$ defeats $\mathcal{A}_2$;
$\mathcal{A}_5$ and $\mathcal{A}_7$ are incomparable (blocking defeaters).

A sequence of arguments called an *argumentation line* thus arises from this attack relation, where each argument defeats its predecessor. To avoid undesirable sequences, which may represent circular argumentation lines, in DeLP an *argumentation line* is *acceptable* if it satisfies certain constraints (see below). A literal L is *warranted* if there exists a non-defeated argument $\mathcal{A}$ supporting L.

Definition 7 ([15]) Let $\Pi_{AM} = (\Theta, \Omega, \Phi, \Delta)$ be a PreDeLP program. Two arguments $\langle \mathcal{A}_1, L_1 \rangle$ and $\langle \mathcal{A}_2, L_2 \rangle$ are *concordant* iff the set $\Theta \cup \Omega \cup \mathcal{A}_1 \cup \mathcal{A}_2$ is non-contradictory.

Definition 8 ([15]) Let Λ be an argumentation line. Λ is an *acceptable argumentation line* iff:

(1) Λ is a finite sequence.
(2) The set Λ_S, of supporting arguments is concordant, and the set Λ_I of interfering arguments is concordant.
(3) No argument $\langle \mathcal{A}_k, L_k \rangle$ in Λ is a subargument of an argument $\langle \mathcal{A}_i, L_i \rangle$ appearing earlier in Λ ($i < k$)
(4) For all i, such that the argument $\langle \mathcal{A}_i, K_i \rangle$ is a blocking defeater for $\langle A_{i-1}, \mathcal{K}_{i-1} \rangle$, if $\langle A_{i+1}, \mathcal{K}_{i+1} \rangle$ exists, then $\langle A_{i+1}, \mathcal{K}_{i+1} \rangle$ is a proper defeater for $\langle A_i, \mathcal{K}_i \rangle$.

Clearly, there can be more than one defeater for a particular argument $\langle \mathcal{A}, L \rangle$. Therefore, many acceptable argumentation lines could arise from $\langle \mathcal{A}, L \rangle$, leading to a tree structure. The tree is built from the set of all argumentation lines rooted in the initial argument. In a dialectical tree, every node (except the root) represents a defeater of its parent, and leaves correspond to undefeated arguments. Each path from the root to a leaf corresponds to a different acceptable argumentation line. A dialectical tree provides a structure for considering all the possible acceptable argumentation lines that can be generated for deciding whether an argument is defeated. This tree is called *dialectical* because it represents an exhaustive dialectical[4] analysis for the argument in its root. For a given argument $\langle \mathcal{A}, L \rangle$, we denote the corresponding dialectical tree as $\mathcal{T}(\langle \mathcal{A}, L \rangle)$.

Given a literal L and an argument $\langle \mathcal{A}, L \rangle$, in order to decide whether or not a literal L is warranted, every node in the dialectical tree $\mathcal{T}(\langle \mathcal{A}, L \rangle)$ is recursively marked as "D" (*defeated*) or "U" (*undefeated*), obtaining a marked dialectical tree $\mathcal{T}^*(\langle \mathcal{A}, L \rangle)$ as follows:

1. All leaves in $\mathcal{T}^*(\langle \mathcal{A}, L \rangle)$ are marked as "U"s, and
2. Let $\langle \mathcal{B}, q \rangle$ be an inner node of $\mathcal{T}^*(\langle \mathcal{A}, L \rangle)$. Then $\langle \mathcal{B}, q \rangle$ will be marked as "U" iff every child of $\langle \mathcal{B}, q \rangle$ is marked as "D". The node $\langle \mathcal{B}, q \rangle$ will be marked as "D" iff it has at least a child marked as "U".

Given an argument $\langle \mathcal{A}, L \rangle$ obtained from Π_{AM}, if the root of $\mathcal{T}^*(\langle \mathcal{A}, L \rangle)$ is marked as "U", then we will say that $\mathcal{T}^*(\langle \mathcal{A}, h \rangle)$ *warrants* L and that L is *warranted* from Π_{AM}. (Warranted arguments correspond to those in the grounded extension of a Dung argumentation system [7].) There is a further requirement when the arguments in the dialectical tree contain presumptions — the conjunction of all presumptions used in even levels of the tree must be consistent. This can give rise to multiple trees for a given literal, as there can potentially be different arguments that make contradictory assumptions.

We can then extend the idea of a dialectical tree to a *dialectical forest*. For a given literal L, a dialectical forest $\mathcal{F}(L)$ consists of the set of dialectical trees for all arguments for L. We shall denote a marked dialectical forest, the set of all marked dialectical trees for arguments for L, as $\mathcal{F}^*(L)$. Hence, for a literal L, we say it is *warranted* if there is at least one argument for that literal in the dialectical forest $\mathcal{F}^*(L)$ that is labeled as "U", *not warranted* if there is at least one argument for the literal $\neg L$ in the dialectical forest $\mathcal{F}^*(\neg L)$ that is labeled as "U", and *undecided* otherwise.

With this, we have a complete description of the analytical model, and can go on to describe the DeLP3E framework.

[4] In the sense of providing reasons for and against a position.

3 The DeLP3E framework

DeLP3E arises from the combination of the environmental model Π_{EM}, and the analytical model Π_{AM}; the two models are held together by the annotation function. This allows elements from the AM to be annotated with elements from the EM. These annotations specify the conditions under which the various statements in the AM can potentially be true.

3.1 Definition

Intuitively, given Π_{AM}, every element of $\Omega \cup \Theta \cup \Delta \cup \Phi$ might only hold in certain worlds in the set $\mathcal{W}_{EM}$ — that is, they are subject to probabilistic events. Therefore, we associate elements of $\Omega \cup \Theta \cup \Delta \cup \Phi$ with a formula from $form_{EM}$. In doing so, we can in turn compute the probabilities of subsets of $\Omega \cup \Theta \cup \Delta \cup \Phi$ using the information contained in Π_{EM}, as we describe shortly. The notion of an *annotation function* associates elements of $\Omega \cup \Theta \cup \Delta \cup \Phi$ with elements of $form_{EM}$.

Definition 9 An *annotation function* is any function $af : \Omega \cup \Theta \cup \Delta \cup \Phi \rightarrow form_{EM}$. We use $[af]$ to denote the set of all annotation functions.

Figure 3 shows an example of an annotation function.

We will sometimes denote annotation functions as sets of pairs $(f, af(f))$ in order to simplify the presentation. Function *af* may come from an expert's knowledge or the data itself. Choosing the correct function and learning the function from data is the topic of ongoing work.

We also note that, by using the annotation function, we may have certain statements that appear as both facts and presumptions (likewise for strict and defeasible rules). However, these constructs would have different annotations, and thus be applicable in different worlds. We note that the annotation function can allow AM facts and strict rules to be true in some EM worlds and false in others – this is why we include facts and strict rules as part of an argument in our framework.

Example 7 Suppose we added the following presumptions to our running example:

$$\phi_3 = l \prec$$
$$\phi_4 = m \prec$$

and suppose we extend *af* as follows:

$$af(\phi_3) = a \wedge b$$
$$af(\phi_4) = a \wedge b \wedge c$$

$af(\theta_{1a}) = af(\theta_{1b})$	$= k \vee \big(f \wedge \big(h \vee (e \wedge l)\big)\big)$	$af(\phi_3)$	$= b$
$af(\theta_2)$	$= i$	$af(\delta_{1a}) = af(\delta_{1b})$	$= \mathsf{True}$
$af(\omega_{1a}) = af(\omega_{1b})$	$= \mathsf{True}$	$af(\delta_2)$	$= \mathsf{True}$
$af(\omega_{2a}) = af(\omega_{2b})$	$= \mathsf{True}$	$af(\delta_3)$	$= \mathsf{True}$
$af(\phi_1)$	$= c \vee a$	$af(\delta_4)$	$= \mathsf{True}$
$af(\phi_2)$	$= f \wedge m$	$af(\delta_{5a}) = af(\delta_{5b})$	$= \mathsf{True}$

Fig. 3 Example annotation function

So, for instance, unlike θ_1, ϕ_3 can potentially be true in any world of the form:

$$\{a, b\}$$

We now have all the components to formally define a DeLP3E program.

Definition 10 Given environmental model Π_{EM}, analytical model Π_{AM}, and annotation function af, a *DeLP3E program* is of the form $\mathcal{I} = (\Pi_{EM}, \Pi_{AM}, af)$. We use notation $[\mathcal{I}]$ to denote the set of all possible programs.

The next step in the definition of DeLP3E is to explore entailment operations. In an entailment query, we are given an AM literal L, probability interval $p \pm \epsilon$, and DeLP3E program $\mathcal{I}$, and we wish to determine if L is entailed by $\mathcal{I}$ with a probability $p \pm \epsilon$. However, before we can formally define this entailment problem, we define a *warranting scenario* to determine the proper environment in question and the entailment bounds (Section 3.2). This is followed by our formal definition and method for computing entailment in Section 3.3.

3.2 Warranting scenario

In DeLP3E, we can consider a world-based approach; that is, the defeat relationship among arguments depends on the current state of the (EM) world.

Definition 11 Let $\mathcal{I} = (\Pi_{EM}, \Pi_{AM}, af)$ be a DeLP3E program, argument $\langle \mathcal{A}, L \rangle$ is *valid* w.r.t. world $w \in \mathcal{W}_{EM}$ iff $\forall c \in \mathcal{A}, w \models af(c)$.

We extend the notion of validity to argumentation lines, dialectical trees, and dialectical forests in the expected way (for instance, an argumentation line is valid w.r.t. w iff all arguments that comprise that line are valid w.r.t. w).

Example 8 Consider worlds $w_1, \ldots, w_8$ from Example 3 along with the argument $\langle \mathcal{A}_5, u \rangle$ from Example 4. This argument is valid in worlds w_1, w_2, w_3, w_4, w_6, and w_7.

We also extend the idea of a dialectical tree w.r.t. worlds; so, for a given world $w \in \mathcal{W}_{EM}$, the dialectical (resp., marked dialectical) tree induced by w is denoted with $\mathcal{T}_w\langle \mathcal{A}, L \rangle$ (resp., $\mathcal{T}^*_w\langle \mathcal{A}, L \rangle$). We require that all arguments and defeaters in these trees be valid with respect to w. Likewise, we extend the notion of dialectical forests in the same manner (denoted with $\mathcal{F}_w(L)$ and $\mathcal{F}^*_w(L)$, resp.). Based on these concepts, we introduce the notion of *warranting scenario*.

Definition 12 Let $\mathcal{I} = (\Pi_{EM}, \Pi_{AM}, af)$ be a DeLP3E program and L be a literal formed with a ground atom from $\mathbf{G}_{AM}$; a world $w \in \mathcal{W}_{EM}$ is said to be a *warranting scenario* for L (denoted $w \vdash_{\mathsf{war}} L$) iff there is a dialectical forest $\mathcal{F}^*_w(L)$ in which L is warranted and $\mathcal{F}^*_w(L)$ is valid w.r.t. w.

We note that for a world w not being a warranting scenario for L, is not the same as being a warranting scenario for $\neg L$. For that we need a dialectical tree $\mathcal{F}^*_w(L')$ in which L' is warranted and $\mathcal{F}^*_w(L')$ is valid w.r.t w where $L' = \neg L$

Example 9 Considering the arguments from Example 8, worlds w_3, w_6, and w_7 are warranting scenarios for argument $\langle \mathcal{A}_5, u \rangle$.

3.3 Entailment in DeLP3E

In this section, we use the idea of a warranting scenario to formally define our entailment problem. We first notice that the set of worlds in the EM where a literal L in the AM *must* be true is exactly the set of warranting scenarios — these are the "necessary" worlds:

$$nec(L) = \{w \in \mathcal{W}_{EM} \mid (w \vdash_{\mathsf{war}} L)\}.$$

Now, the set of worlds in the EM where AM literal L *can* be true is the following — these are the "possible" worlds:

$$poss(L) = \{w \in \mathcal{W}_{EM} \mid w \nvdash_{\mathsf{war}} \neg L\}.$$

Example 10 Following from Example 8, we have that:

$nec(u) = \{w_3, w_6, w_7\}$ and $poss(u) = \{w_1, w_2, w_3, w_4, w_6, w_7\}$.

Definition 13 We define $for(w) = \bigwedge_{a \in w} a \wedge \bigwedge_{a \notin w} \neg a$, which denotes the *formula* that has w as its only model. Also, we extend this notation to sets of words: $for(W) = \bigvee_{w \in W} for(w)$.

Definition 14 (Entailment) Given DeLP3E program, $\mathcal{I} = (\Pi_{EM}, \Pi_{AM}, af)$, AM literal L and probability interval $p \pm \epsilon$, we say that $\mathcal{I}$ entails L with probability $p \pm \epsilon$ iff all probability distributions Pr that satisfy Π_{EM} satisfy $for(nec(L)) : p \pm \epsilon$ and $for(poss(L)) : p \pm \epsilon$.

We will also refer to the tightest bound $[p - \epsilon, p + \epsilon]$ such that $\mathcal{I}$ entails L with a probability $p \pm \epsilon$ as the "tightest entailment bounds." The intuition behind the above definition of entailment is as follows. Let ℓ be the maximum value for $p - \epsilon$ and u be the minimum value for $p + \epsilon$ before we can no longer say that $\mathcal{I}$ entails L with probability $p \pm \epsilon$. In this case, we can define probability distributions Pr^-_{poss}, Pr^+_{poss}, Pr^-_{nec}, Pr^+_{nec} as follows:

- Pr^-_{poss} satisfies Π_{EM} and assigns the smallest possible probability to worlds in $for(poss(L))$.
- Pr^+_{poss} satisfies Π_{EM} and assigns the largest possible probability to worlds in $for(poss(L))$.
- Pr^-_{nec} satisfies Π_{EM} and assigns the smallest possible probability to worlds in $for(nec(L))$.
- Pr^+_{nec} satisfies Π_{EM} and assigns the largest possible probability to worlds $for(nec(L))$.

We only need to compare $Pr^-_{poss}(poss(L))$ and $Pr^-_{nec}(nec(L))$ for finding the lower bound since $Pr^+_{poss}(poss(L)) \geq Pr^-_{poss}(poss(L))$ and $Pr^+_{nec}(nec(L)) \geq Pr^-_{nec}(nec(L))$. Similar reasoning holds for the case of finding the upper bound. Thus, we get the following relationships:

$$\ell = \min\left(Pr^-_{poss}(poss(L)), Pr^-_{nec}(nec(L))\right) \quad (1)$$

$$u = \max\left(Pr^+_{poss}(poss(L)), Pr^+_{nec}(nec(L))\right) \quad (2)$$

However, we note that as $nec(L) \subseteq poss(L)$ we have the following:

$$\ell = Pr^{-}_{nec}(nec(L)) \tag{3}$$

$$u = Pr^{+}_{poss}(poss(L)) \tag{4}$$

We note that (2) and (4) is equivalent to the belief and plausibility values of L defined in the *Dempster-Shafer theory* [38].

Hence, the tightest possible entailment bounds that can be assigned to a literal can be no less than the lower bound of the probability assigned to the necessary warranting scenarios and no more than the probability assigned to the possible warranting scenarios. Hence, we can compute the tightest probability bound such that L is entailed (denoted $\mathbf{P}_{L,Pr,\mathcal{I}}$) as follows:

$$\ell_{L,Pr,\mathcal{I}} = \sum_{w \in nec(L)} Pr^{-}_{nec(w)}, \quad u_{L,Pr,\mathcal{I}} = \sum_{w \in poss(L)} Pr^{+}_{poss(w)}$$

$$\ell_{L,Pr,\mathcal{I}} \leq \mathbf{P}_{L,Pr,\mathcal{I}} \leq u_{L,Pr,\mathcal{I}}$$

Thus, in interval form we have:

$$\mathbf{P}_{L,Pr,\mathcal{I}} = \left(\ell_{L,Pr,\mathcal{I}} + \frac{u_{L,Pr,\mathcal{I}} - \ell_{L,Pr,\mathcal{I}}}{2}\right) \pm \frac{u_{L,Pr,\mathcal{I}} - \ell_{L,Pr,\mathcal{I}}}{2}.$$

Now let us consider the computation of tightest probability bounds for entailment on a literal when we are given a knowledge base $\mathcal{K}_{EM}$ in the environmental model, which is specified in $\mathcal{I}$, instead of a probability distribution over all worlds. For a given world $w \in \mathcal{W}_{EM}$, let $for(w) = \left(\bigwedge_{a \in w} a\right) \wedge \left(\bigwedge_{a \notin w} \neg a\right)$ — that is, a formula that is satisfied only by world w. Now we can determine the upper and lower bounds on the probability of a literal w.r.t. $\mathcal{K}_{EM}$ (denoted $\mathbf{P}_{L,\mathcal{I}}$) as follows:

$$\ell_{L,\mathcal{I}} = \text{EP-LP-MIN}\left(\mathcal{K}_{EM}, \bigvee_{w \in nec(L)} for(w)\right)$$

$$u_{L,\mathcal{I}} = \text{EP-LP-MAX}\left(\mathcal{K}_{EM}, \bigvee_{w \in poss(L)} for(w)\right)$$

$$\ell_{L,\mathcal{I}} \leq \mathbf{P}_{L,\mathcal{I}} \leq u_{L,\mathcal{I}}$$

Hence, $\mathbf{P}_{L,\mathcal{I}} = \left(\ell_{L,\mathcal{K}_{EM}} + \frac{u_{L,\mathcal{I}} - \ell_{L,\mathcal{I}}}{2}\right) \pm \frac{u_{L,\mathcal{I}} - \ell_{L,\mathcal{I}}}{2}$.

Example 11 Consider argument $\langle \mathcal{A}_5, u \rangle$ from Example 8. We can compute $\mathbf{P}_{u,\mathcal{I}}$ (where $\mathcal{I} = (\Pi'_{EM}, \Pi_{AM}, af)$).

Note that for the upper bound, the linear program we need to set up is the one shown in Example 3. For the lower bound, the objective function changes to: $\min x_3 + x_6 + x_7$. From these linear constraints, we obtain:

$$\mathbf{P}_u = 0.7 \pm 0.2$$

In the following, we study the problem of consistency in our framework, which is the basis of the belief revision operators studied later on.

4 Belief revision in DeLP3E programs

Even though our framework relies heavily on argumentation and reasoning under uncertainty, inconsistency in our knowledge base can still arise. For instance, the knowledge encoded in the environmental model could become contradictory, which would preclude any probability distribution from satisfying that part of the knowledge base. Even on the argumentation side, despite that fact that argumentation formalisms in general are inconsistency tolerant, there may be problems with inconsistency. For example, it would be problematic for DeLP3E if the set of strict facts and strict rules were contradictory, *and* the set of contradictory elements all arise under the same environmental conditions.

4.1 Consistency of DeLP3E programs

In this section, we first explore what forms of inconsistency can arise in DeLP3E programs before going on to examine in detail how ideas from belief revision can be applied to deal with this inconsistency. We use the following notion of (classical) consistency of PreDeLP programs: Π is said to be *consistent* if there does not exist a ground literal a s.t. $\Pi \vdash a$ and $\Pi \vdash \neg a$. For DeLP3E programs, there are two main kinds of inconsistency that can be present; the first is what we refer to as EM, or Type I, (in)consistency.

Definition 15 An environmental model Π_{EM} is *Type I consistent* iff there exists a probability distribution Pr over the set of worlds $\mathcal{W}_{EM}$ that satisfies Π_{EM}.

We illustrate this type of consistency in the following example.

Example 12 It is possible to create probabilistic knowledge bases for which there is no satisfying probability distribution. The following formula is a simple example of such a case:

$$rain \vee hail : 0.3 \pm 0;$$
$$rain \wedge hail : 0.5 \pm 0.1.$$

The above is an example of Type I inconsistency in DeLP3E, as it arises from the fact that there is no satisfying interpretation for the EM knowledge base.

However, even if the EM is consistent, the interaction between the annotation function and facts and strict rules can still cause another type of inconsistency to arise. We will refer to this as combined, or Type II, (in)consistency.

Definition 16 A DeLP3E program $\mathcal{I} = (\Pi_{EM}, \Pi_{AM}, af)$, with $\Pi_{AM} = \langle \Theta, \Omega, \Phi, \Delta \rangle$, is *Type II consistent* iff: given any probability distribution Pr that satisfies Π_{EM}, if there exists a world $w \in \mathcal{W}_{EM}$ such that $\bigcup_{x \in \Theta \cup \Omega \mid w \models af(x)} \{x\}$ is inconsistent, then we have $Pr(w) = 0$.

Thus, any EM world in which the set of associated facts and strict rules are inconsistent (we refer to this as "classical consistency") must always be assigned a zero probability. The intuition is as follows: any subset of facts and strict rules are thought to be true under certain circumstances — these circumstances are determined through the annotation function and can be expressed as sets of EM worlds. Suppose there is a world where two contradictory facts can both be considered to be true (based on the annotation function). If this occurs, then

there must not exist a probability distribution that satisfies the program Π_{EM} that assigns such a world a non-zero probability, as this world leads to an inconsistency. We provide a more concrete example of Type II inconsistency next.

Example 13 Consider the environmental model from Example 2 (Page 8), the analytical model shown in Fig. 1 (Page 10), and the annotation function shown in Figure 3 (Page 15). Suppose the following fact is added to the argumentation model:

$$\theta_3 = \neg p,$$

and that the annotation function is expanded as follows:

$$af(\theta_3) = k \wedge \neg f$$

Clearly, fact θ_3 is in direct conflict with fact θ_{1a}. However, this does not necessarily mean that there is an inconsistency. For instance, by the annotation function, θ_{1a} holds in the world $\{k, f\}$ while θ_3 does not. However, let's consider following world $w = \{k\}$. Note that $w \models af(\theta_3)$ and $w \models af(\theta_2)$. Hence, in this world both contradictory facts can occur. However, can this world be assigned a non-zero probability? A simple examination of the environmental model indicates that it can. Hence, in this case, we have Type II inconsistency.

We say that a DeLP3E program is *consistent* iff it is both Type I and Type II consistent. However, in this paper, we focus on Type II consistency and assume that the program is Type I consistent. Figure 4 gives a straightforward approach to identifying Type II inconsistent DeLP3E programs by running breath-first search on a set of $\Theta \cup \Omega$. The algorithm works by examining all subsets of a set of facts and strict rules to find inconsistent subsets whose corresponding formula in the environmental model can be assigned a non-zero probability. The following result states its correctness.

Proposition 1 *For Type I consistent DeLP3E program $\mathcal{I} = (\Pi_{EM}, \Pi_{AM}, af)$ where Θ and Ω are the sets of facts and strict rules in Π_{AM}, then CON-CHK-BFS($\Pi_{EM}, \Pi_{AM}, af, d, \{\Theta \cup \Omega\}$) (where $d = |\Theta \cup \Omega|$) returns INCONSISTENT iff the DeLP3E is Type II inconsistent.*

Proof The algorithm takes the set of facts and strict rules and checks the consistency of it by checking the value of the probability distribution on the set and the subsets of the given set. If in any step there exists a subset of facts and strict rules that is not Type I consistent, the algorithm checks the value of the probability distribution; if it is not zero, it will return INCONSISTENT. BWOC, suppose the algorithm has returned INCONSISTENT for a DeLP3E program that is consistent. So, there exist a subset S of size d of facts and

Algorithm CON-CHK-BFS($\Pi_{EM}, \Pi_{AM}, af, d, \mathcal{S} = \{S_1, \ldots, S_n\}$)

1. $\mathcal{S}' := \emptyset$
2. For each $S_i \in \mathcal{S}$ where S_i is not classically consistent, do the following:
   a. If Pr is such that $Pr \models \Pi_{EM}$ and $Pr(\bigwedge_{s \in S_i} af(s)) > 0$ then return INCONSISTENT and terminate;
   b. Else, $\mathcal{S}' := \mathcal{S}' \cup \{S' \subseteq S_i \mid |S'| = |S_i| - 1\}$;
3. If $d = 1$ return CONSISTENT;
4. Else, return CON-CHK-BFS($\Pi_{EM}, \Pi_{AM}, af, d - 1, \mathcal{S}'$).

Fig. 4 A straightforward BFS-based algorithm for consistency checking

strict rules for which the algorithm has returned INCONSISTENT, while S is consistent. Because the algorithm has returned INCONSISTENT, the set S is classically inconsistent. It also means that $\nexists\ w \in W_{AM}\ s.t.\ w \models \bigwedge_{s \in S}\{s\}$ and $\exists\ Pr\ s.t.\ Pr(af(s)) > 0$. This is in contradiction with the assumption of consistency of S. For the other direction, consider a DeLP3E program that is inconsistent. Since the program is inconsistent, there exists a world $w \in \mathcal{W}_{EM}$ such that $\bigcup_{x \in \Theta \cup \Omega \mid w \models af(x)}\{x\}$ is inconsistent and $Pr(w) > 0$. Since $\bigcup_{x \in \Theta \cup \Omega \mid w \models af(x)}\{x\}$ is a subset of the facts and rules, the algorithm checks its consistency in some iteration. Since the subset is inconsistent and the probability value assigned to it is greater than zero, the algorithm returns INCONSISTENT. □

However, we note that even with an oracle for checking the classical consistency of a subset (line 2) and for determining the upper bound on the probability of the annotations (line 2a), this algorithm is still intractable as it explores all subsets of $\Theta \cup \Omega$. One possible way to attack this intractability is to restrict the depth of the search by setting d to be less than the size of $\Theta \cup \Omega$. In this case, we get the following result:

Proposition 2 *Given Type I consistent DeLP3E program $\mathcal{I} = (\Pi_{EM}, \Pi_{AM}, af)$, where Θ and Ω are the sets of facts and strict rules in Π_{AM} and $d < |\Theta \cup \Omega|$, then if CON-CHK-BFS($\Pi_{EM}, \Pi_{AM}, af, d, \{\Theta \cup \Omega\}$) returns INCONSISTENT, the program $\mathcal{I}$ is Type II inconsistent.*

Proof Suppose, BWOC that CON-CHK-BFS($\Pi_{EM}, \Pi_{AM}, af, d, \{\Theta \cup \Omega\}$) returns INCONSISTENT, and the program $\mathcal{I}$ is Type II consistent. We claim that by showing that, under the condition of the statement, that if $\mathcal{I}$ is Type II consistent, then CON-CHK-BFS($\Pi_{EM}, \Pi_{AM}, af, d, \{\Theta \cup \Omega\}$) must return CONSISTENT (giving a contradiction). This is due to the following: since calling CON-CHK-BFS($\Pi_{EM}, \Pi_{AM}, af, d, \{\Theta \cup \Omega\}$ where $d = |\Theta \cup \Omega|$ returns CONSISTENT when the program is consistent, then the algorithm returns CONSISTENT for every subset of $\Theta \cup \Omega$ smaller than itself. As a result, CON-CHK-BFS($\Pi_{EM}, \Pi_{AM}, af, d, \{\Theta \cup \Omega\}$) (where $d < |\Theta \cup \Omega|$) returns CONSISTENT. □

Therefore, by restricting depth, we can view this algorithm as an "anytime" approach, essentially searching for a world leading to an inconsistent program and not halting until it does.

In the following sections, we explore three methods for resolving Type II inconsistency through belief revision. We summarize them briefly below.

Revise the EM. The probabilistic model can be changed in order to force the worlds that induce contradicting strict knowledge to have probability zero. In general, this type of revision by itself is not ideal as it will not work in all cases. We discuss this method in Section 4.2.

Revise the AM. The argumentation model can be changed in such a way that the set of strict rules and facts is consistent. If this is the case, then Type II consistency follows. We discuss this method in Section 4.3.

Revise the annotation function. The annotations involved in the inconsistency can be changed so that the conflicting information in the AM does not become induced under any possible world. This can be viewed as a generalization of AM revision. We discuss this method in Section 4.4.

4.2 EM-based belief revision

We now study belief revision through updating the environmental model only (Π_{EM}). Suppose that Π_{EM} is consistent, but that the overall program is Type II inconsistent. Then, there must exist a set of worlds in the EM such that there exists a probability distribution that assigns each of them a non-zero probability. This gives rise to the following result.

Proposition 3 *If there exists a probability distribution* Pr *that satisfies* Π_{EM} *s.t. there exists a world* $w \in \mathcal{W}_{EM}$ *where* $Pr(w) > 0$ *and* $\bigcup_{x \in \Theta \cup \Omega \mid w \models af(x)} \{x\}$ *is inconsistent (Type II inconsistency), then any change made in order to resolve this inconsistency by modifying only* Π_{EM} *yields a new EM* Π'_{EM} *such that* $\left(\bigwedge_{a \in w} a \wedge \bigwedge_{a \notin w} \neg a \right) : 0 \pm 0$ *is entailed by* Π'_{EM}.

Proof Suppose by contradiction that $\Pi'_{EM} \not\models \left(\bigwedge_{a \in w} a \wedge \bigwedge_{a \notin w} \neg a \right) : 0 \pm 0$. By hypothesis, we have that $\bigcup_{x \in \Theta \cup \Omega \mid w \models af(x)} \{x\}$ is inconsistent and the changes made to Π_{EM} resolve this inconsistency. Therefore, according to Definition 16, $Pr(w) = 0$, which is equivalent to the condition $\Pi'_{EM} \models \left(\bigwedge_{a \in w} a \wedge \bigwedge_{a \notin w} \neg a \right) : 0 \pm 0$. □

Proposition 3 seems to imply an easy strategy to resolve Type II inconsistencies: add formulas to Π_{EM} forcing the necessary worlds to have a zero probability. However, this may lead to Type I inconsistencies in the resulting model Π'_{EM}. If we are applying an EM-only strategy to resolve inconsistencies, this would then lead to further adjustments to Π'_{EM} in order to restore Type I consistency. We illustrate this situation in the following example.

Example 14 Consider two contradictory facts in an AM: a and $\neg a$ such that $af(a) = p$ and $af(\neg a) = q$. Suppose that p and q are the only atoms in the EM, and that we have:

$$\begin{aligned} p &: 0.4 \pm 0 \\ q &: 0.8 \pm 0.1 \\ \neg p \wedge \neg q &: 0.2 \pm 0.1 \end{aligned}$$

which is consistent since the following distribution satisfies all constraints:

$Pr(\{p\}) = 0.2;$
$Pr(\{p, q\}) = 0.2;$
$Pr(\{q\}) = 0.5;$
$Pr(\{\}) = 0.1.$

Now, to restore Type II consistency of our simple DeLP3E program, we can add formula $p \wedge q : 0 \pm 0$ to the EM so that world $\{p, q\}$ is forced to have probability zero. However, this leads to another inconsistency, this time of Type I, since putting together all the constraints we have:

$Pr(\{p, q\}) = 0;$
$Pr(\{p\}) + Pr(\{p, q\}) = 0.4;$
$Pr(\{q\}) + Pr(\{p, q\}) = 0.8 \pm 0.1;$
$Pr(\{\}) = 0.2 \pm 0.1;$
$Pr(\{p\}) + Pr(\{p, q\}) + Pr(\{q\}) + Pr(\{\}) = 1;$

which is clearly inconsistent. Repairing this inconsistency involves changing the EM further, for instance by relaxing the bounds in the first two formulas to accommodate the probability mass that world $\{p, q\}$ had before and can no longer hold.

In the previous example, we saw how changes made to repair Type II inconsistencies could lead to Type I inconsistencies. It is also possible that changing Π'_{EM} (for instance, by removing elements, relaxing probability bounds of the sentences, etc.) causes Type II inconsistency in the overall DeLP3E program — this would lead to the need to set more EM worlds to a probability of zero. Unfortunately, this process is not guaranteed to arrive at a fully consistent program before being unable to continue; consider the following example, where the process cannot even begin.

Example 15 Consider an AM composed of several contradictory facts and an EM with just two atoms, as in the previous example, and the following annotation function:

$$\begin{array}{llll} af(a) = p & af(b) = \neg p & af(c) = \neg p & af(d) = q \\ af(\neg a) = q & af(\neg b) = \neg q & af(\neg c) = p & af(\neg d) = \neg q \end{array}$$

Modifying the EM so that no two contradictory literals ever hold at once in a world that has a non-zero probability leads to the constraints:

$Pr(\{p, q\}) = 0;$
$Pr(\{p\}) = 0;$
$Pr(\{q\}) = 0;$
$Pr(\{\}) = 0;$
$Pr(\{p\}) + Pr(\{p, q\}) + Pr(\{q\}) + Pr(\{\}) = 1;$

As in the previous example, the probability mass cannot be accommodated within these constraints. It would thus be impossible to restore consistency by only modifying Π_{EM}.

We thus arrive at the following observation from Example 15:

Observation 1 *Given a Type II inconsistent DeLP3E program, consistency cannot always be restored via modifications to* Π_{EM} *alone.*

Therefore, due to this line of reasoning, in this paper we focus our efforts on modifications to the other two components of a DeLP3E framework: the AM and the annotation function, as described in the next two sections. Approaches combining two or more of these methods are the topic of future work.

4.3 AM-based belief revision

The result of the previous section indicates that EM-based belief revision of a DeLP3E framework (at least by itself) is not a tenable solution. Hence, in this section, we resort to an alternate approach in which we only modify the AM (Π_{AM}). In this section (and the next), given a DeLP3E program $\mathcal{I} = (\Pi_{EM}, \Pi_{AM}, af)$, with $\Pi_{AM} = \Omega \cup \Theta \cup \Delta \cup \Phi$, we are interested in solving the problem of incorporating an epistemic input (f, af') into $\mathcal{I}$, where f is either an atom or a rule and af' is equivalent to af, except for its expansion to include f. For ease of presentation, we assume that f is to be incorporated as a fact or strict rule, as incorporating defeasible knowledge can never lead to inconsistency since any contradicting presumption can be defeated by another, and hence presumptions can rule out each other. As we are only conducting Π_{AM} revisions, for $\mathcal{I} = (\Pi_{EM}, \Pi_{AM}, af)$ and input (f, af') we denote the revision as follows: $\mathcal{I} \bullet (f, af') = (\Pi_{EM}, \Pi'_{AM}, af')$ where Π'_{AM} is the revised

argumentation model. We also slightly abuse notation for the sake of presentation, as well as introduce notation to convert sets of worlds to/from formulas:

- $\mathcal{I} \cup (f, af')$ to denote $\mathcal{I}' = (\Pi_{EM}, \Pi_{AM} \cup \{f\}, af')$.
- $(f, af') \in \mathcal{I} = (\Pi_{AM}, \Pi_{EM}, af)$ to denote $f \in \Pi_{AM}$ and $af = af'$.
- $\mathcal{W}^0_{EM}(\mathcal{I}) = \{w \in \mathcal{W}_{EM} \mid \Pi^{\mathcal{I}}_{AM}(w)$ is inconsistent$\}$
- $\mathcal{W}^I_{EM}(\mathcal{I}) = \{w \in \mathcal{W}^0_{EM} \mid \exists Pr$ *s.t.* $Pr \models \Pi_{EM} \wedge Pr(w) > 0\}$

Intuitively, the set $\mathcal{W}^0_{EM}(\mathcal{I})$ contains all the EM worlds for a given program where the corresponding knowledge base in the AM is classically inconsistent and $\mathcal{W}^I_{EM}(\mathcal{I})$ is a subset of these that can be assigned a non-zero probability — the latter are the worlds where inconsistency in the AM can arise.

4.3.1 Postulates for AM-based belief revision

We now analyze the rationality postulates for non-prioritized revision of belief bases first introduced in [20] and generalized in [10], in the context of AM-based belief revision of DeLP3E programs.

AM inclusion For $\mathcal{I} \bullet (f, af') = (\Pi_{EM}, \Pi'_{AM}, af')$, $\Pi'_{AM} \subseteq \Pi_{AM} \cup \{f\}$.
This postulate states that the revised AM knowledge base is a subset of the union of the original AM knowledge base and the input.

AM vacuity If $\mathcal{I} \cup (f, af')$ is consistent, then $\mathcal{I} \bullet (f, af') \subseteq \mathcal{I} \cup (f, af')$.
If simply adding the input does not cause inconsistency, then the revision operator does precisely that.

AM consistency preservation If $\mathcal{I}$ is consistent, then $\mathcal{I} \bullet (f, af')$ is also consistent.
The operator maintains a consistent program.

AM weak success If $\mathcal{I} \cup (f, af')$ is consistent, then $(f, af') \in \mathcal{I} \bullet (f, af')$.
Whenever the simple addition of the input does not cause inconsistencies to arise, the result will contain the input.

If a portion of the AM knowledge base is removed by the operator, then there exists a subset of the remaining knowledge base that is not consistent with the removed element and f.

AM pertinence For $\mathcal{I} \bullet (f, af') = (\Pi_{EM}, \Pi'_{AM}, af')$, where $\Pi'_{AM} = \Theta' \cup \Omega' \cup \Phi' \cup \Delta'$, for each $g \in \Theta \cup \Omega \setminus \Pi'_{AM}$ there exists $Y_g \supseteq \Theta' \cup \Omega' \cup \{f\}$ s.t. Y_g is consistent and $Y_g \cup \{g\}$ is inconsistent.
If a portion of the AM knowledge base is removed by the operator, then there exists a superset of the remaining knowledge base that is not consistent with the removed element and f.

AM uniformity 1 Let $(f, af'_1), (g, af'_2)$ be two inputs where $\mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af'_1)) = \mathcal{W}^I_{EM}(\mathcal{I} \cup (g, af'_2))$; for all $X \subseteq \Theta \cup \Omega$; if $X \cup \{f\}$ is inconsistent iff $X \cup \{g\}$ is inconsistent, then $\Theta'_1 \cup \Omega'_1 \setminus \{f\} = \Theta'_2 \cup \Omega'_2 \setminus \{g\}$ where $\mathcal{I} \bullet (f, af'_1) = (\Pi_{EM}, \Pi_{AM'_1}, af'_1)$ and $\mathcal{I} \bullet (g, af'_2) = (\Pi_{EM}, \Pi_{AM'_2}, af'_2)$ and $\Pi_{AM'_i} = \Theta'_i \cup \Omega'_i \cup \Phi'_i \cup \Delta'_i$.
If two inputs result in the same set of EM worlds leading to inconsistencies in an AM knowledge base, and the consistency between analogous subsets (when joined with the

respective input) are the same, then the remaining elements in the AM knowledge base are the same.

AM uniformity 2 Let $(f, af'_1), (g, af'_2)$ be two inputs where $\mathcal{W}^{\mathcal{I}}_{EM}(\mathcal{I} \cup (f, af'_1)) = \mathcal{W}^{\mathcal{I}}_{EM}(\mathcal{I} \cup (g, af'_2))$; for all $X \subseteq \Theta \cup \Omega$; if $X \cup \{f\}$ is inconsistent iff $X \cup \{g\}$ is inconsistent, then $(\Theta \cup \Omega) \setminus (\Theta'_1 \cup \Omega'_1) = (\Theta \cup \Omega) \setminus (\Theta'_2 \cup \Omega'_2)$ where $\mathcal{I} \bullet (f, af'_1) = (\Pi_{EM}, \Pi_{AM'_1}, af'_1)$ and $\mathcal{I} \bullet (g, af'_2) = (\Pi_{EM}, \Pi_{AM'_2}, af'_2)$ and $\Pi_{AM'_i} = \Theta'_i \cup \Omega'_i \cup \Phi'_i \cup \Delta'_i$.
If two inputs result in the same set of EM worlds leading to inconsistencies in an AM knowledge base, and the consistency between analogous subsets (when joined with the respective input) are the same, then the removed elements in the AM knowledge base are the same.

We can show an equivalence between the Uniformity postulates under certain conditions.

Proposition 4 *For operator $\bullet$ where for program $\mathcal{I} \bullet (f, af') = (\Pi_{EM}, \Pi'_{AM}, af')$ and $\Pi'_{AM} \subseteq \Pi_{AM} \cup \{f\}$, we have that $\bullet$ satisfies AM Uniformity 1 iff it also satisfies AM Uniformity 2.*

Proof (If) Suppose BWOC that $\bullet$ satisfies AM Uniformity 1 and does not satisfy AM Uniformity 2. Then (for the two inputs as specified by the Uniformity postulates) $(\Theta \cup \Omega) \setminus (\Theta'_1 \cup \Omega'_1) = (\Theta \cup \Omega) \setminus (\Theta'_2 \cup \Omega'_2)$ and $\Theta'_1 \cup \Omega'_1 \setminus \{f\} \neq \Theta'_2 \cup \Omega'_2 \setminus \{g\}$. However, this is equivalent to $(\Theta \cup \Omega) \setminus (\Theta'_1 \cup \Omega'_1) = (\Theta \cup \Omega) \setminus (\Theta'_2 \cup \Omega'_2)$ — hence, we arrive at a contradiction.
(Only-If) Mirrors the above claim. □

4.3.2 AM-based revision operators

In this section, we define a class of operators that satisfies all of the AM rationality postulates of the previous section. We also show that there are no operators outside this class that satisfy all of the postulates.

First, we introduce notation $CandPgm_{AM}(\mathcal{I})$, which denotes a set of maximal consistent subsets of Π_{AM}. So, if $\mathcal{I}$ is consistent, then $CandPgm_{AM}(\mathcal{I}) = \{\Pi_{AM}\}$.

$$\begin{aligned} CandPgm_{AM}(\mathcal{I}) = \{&\Pi'_{AM} \mid \Pi'_{AM} \subseteq \Theta \cup \Omega \text{ s.t. } \Pi'_{AM} \text{ is consistent and} \\ &\nexists \Pi''_{AM} \subseteq \Theta \cup \Omega \text{ s.t. } \Pi''_{AM} \supset \Pi'_{AM} \text{ s.t. } \Pi''_{AM} \text{ is consistent}\} \end{aligned}$$

For our first result, we show that an operator returning any subset of an element of $CandPgm_{AM}(\mathcal{I})$ is a necessary and sufficient condition for satisfying both the Inclusion and Consistency Preservation postulates.

Lemma 1 *Given program $\mathcal{I}$ and input (f, af'), operator $\bullet$ satisfies Inclusion and Consistency Preservation iff for $\mathcal{I} \bullet (f, af') = (\Pi_{EM}, \Pi'_{AM}, af')$, there exists an element $X \in CandPgm_{AM}(\mathcal{I} \cup (f, af'))$ s.t. $(\Theta \cup \Omega \cup \{f\}) \cap \Pi'_{AM} \subseteq X$.*

Proof (If) Suppose, BWOC, that there exists $X \in CandPgm_{AM}(\mathcal{I} \cup (f, af'))$ s.t. $(\Theta \cup \Omega \cup \{f\}) \cap \Pi'_{AM} \subseteq X$, but either Inclusion or Consistency Preservation is not satisfied. However, the elements of $CandPgm_{AM}(\mathcal{I} \cup (f, af'))$ are all classically consistent with all subsets of $\Pi_{AM} \cup \{f\}$, which is a contradiction.

(Only-If) Suppose, BWOC, that the operator satisfies both Inclusion and Consistency Preservation and there does not exist $X \in Cand Pgm_{AM}(\mathcal{I} \cup (f, af'))$ s.t. $(\Theta \cup \Omega \cup \{f\}) \cap \Pi'_{AM} \subseteq X$. Then, $(\Theta \cup \Omega \cup \{f\}) \cap \Pi_{AM}$ is a subset of $\Pi_{AM} \cup \{f\}$ and $(\Theta \cup \Omega \cup \{f\}) \cap \Pi'_{AM}$ is classically consistent.

However, by definition, this would mean that it must also be a subset of an element in $Cand Pgm_{AM}(\mathcal{I} \cup (f, af'))$. □

Our next result extends Lemma 1 by showing that elements of $\Pi_{AM} \cup \{f\}$ that are retained are also elements of $Cand Pgm_{AM}(\mathcal{I} \cup (f, af'))$ if and only if the operator satisfies Inclusion, Consistency Preservation, and Pertinence (simultaneously).

Lemma 2 *Given program $\mathcal{I}$ and input (f, af'), operator $\bullet$ satisfies Inclusion, Consistency Preservation, and Pertinence iff for $\mathcal{I} \bullet (f, af') = (\Pi_{EM}, \Pi'_{AM}, af')$, we have $(\Theta \cup \Omega \cup \{f\}) \cap \Pi'_{AM} \in Cand Pgm_{AM}(\mathcal{I} \cup (f, af'))$.*

Proof (If) Suppose, BWOC, that $(\Theta \cup \Omega \cup \{f\}) \cap \Pi'_{AM} \in Cand Pgm_{AM}(\mathcal{I} \cup (f, af'))$, (which, by Lemma 1, satisfies both Consistency and Inclusion) but does not satisfy Pertinence. As $|\Theta \cup \Omega \setminus X| > 0$, then $f \in (\Theta \cup \Omega \cup \{f\}) \cap \Pi'_{AM}$. This means that $(\Theta \cup \Omega \cup \{f\}) \cap \Pi'_{AM} \supseteq X \cup \{f\}$, which also yields a contradiction.
(Only-If) Suppose, BWOC, that the operator satisfies Inclusion, Consistency Preservation, and Pertinence but $(\Theta \cup \Omega \cup \{f\}) \cap \Pi'_{AM} \notin Cand Pgm_{AM}(\mathcal{I} \cup (f, af'))$. As the operator satisfies Pertinence, and by Lemma 1, $(\Theta \cup \Omega \cup \{f\}) \cap \Pi'_{AM} \in V = \{X \mid \exists Y \in Cand Pgm_{AM}(\mathcal{I} \cup (f, af')) \wedge X \supseteq Y\}$. As $(\Theta \cup \Omega \cup \{f\}) \cap \Pi'_{AM} \notin Cand Pgm_{AM}(\mathcal{I} \cup (f, af'))$, we have $(\Theta \cup \Omega \cup \{f\}) \cap \Pi'_{AM} \in Z = \{X \mid \exists Y \in Cand Pgm_{AM}(\mathcal{I} \cup (f, af')) \wedge X \supset Y\}$. However, this violates Lemma 1 — we have thus arrived at a contradiction. □

To support the satisfaction of the first Uniformity postulate, we provide the following lemma that shows for a consistent program where two inputs cause inconsistencies to arise in the same way, that the set of candidate replacement programs (minus the added AM formula) is the same.

Lemma 3 *Let $\mathcal{I} = (\Pi_{EM}, \Pi_{AM}, af)$ be a consistent program, (f_1, af'_1), (f_2, af'_2) be two inputs, and $\mathcal{I}_i = (\Pi_{EM}, \Pi_{AM} \cup \{f_i\}, af'_i)$. If $\mathcal{W}^I_{EM}(\mathcal{I}_1) = \mathcal{W}^I_{EM}(\mathcal{I}_2)$, then for all $X \subseteq \Theta \cup \Omega$ we have that:*

1. *If $X \cup \{f_1\}$ is inconsistent $\Leftrightarrow$ $X \cup \{f_2\}$ is inconsistent, then:*
   $\{X \setminus \{f_1\} \mid X \in Cand Pgm_{AM}(\mathcal{I}_1)\} = \{X \setminus \{f_2\} \mid X \in Cand Pgm_{AM}(\mathcal{I}_2)\}$.
2. *If $\{X \setminus \{f_1\} \mid X \in Cand Pgm_{AM}(\mathcal{I}_1)\} = \{X \setminus \{f_2\} \mid X \in Cand Pgm_{AM}(\mathcal{I}_2)\}$ then $X \cup \{f_1\}$ is inconsistent $\Leftrightarrow$ $X \cup \{f_2\}$ is inconsistent.*

Proof (If) Suppose BWOC that for all $X \subseteq \Theta \cup \Omega$ we have that $X \cup \{f_1\}$ is inconsistent iff $X \cup \{f_2\}$ is inconsistent, but $\{X \setminus \{f_1\} \mid X \in Cand Pgm_{AM}(\mathcal{I}_1)\} \neq \{X \setminus \{f_2\} \mid X \in Cand Pgm_{AM}(\mathcal{I}_2)\}$. However, the pre-condition of this statement implies that $\{X \setminus \{f_1\} \mid X \subseteq Cand Pgm_{AM}(\mathcal{I}_1)\} = \{X \setminus \{f_2\} \mid X \subseteq Cand Pgm_{AM}(\mathcal{I}_2)$, which gives us a contradiction.
(Only-If) Suppose BWOC that $\{X \setminus \{f_1\} \mid X \in Cand Pgm_{AM}(\mathcal{I}_1)\} = \{X \setminus \{f_2\} \mid X \in Cand Pgm_{AM}(\mathcal{I}_2)\}$, but there exists a set $X \subseteq \Theta \cup \Omega$ s.t. exactly one of $X \cup \{f_1\}$, $X \cup \{f_2\}$ is inconsistent. As a first case, let us assume that $\Theta \cup \Omega \cup \{f_1\}$ is consistent. As $\{X \setminus \{f_1\} \mid X \in Cand Pgm_{AM}(\mathcal{I}_1)\} = \{X \setminus \{f_2\} \mid X \in Cand Pgm_{AM}(\mathcal{I}_2)\}$, this implies

that $\Theta \cup \Omega \cup \{f_2\}$ must also be consistent as each of those sets must then have exactly one element. In this case, a contradiction arises, hence both $\Theta \cup \Omega \cup \{f_1\}$ and $\Theta \cup \Omega \cup \{f_2\}$ must be classically inconsistent. Now let us consider the other case. As $\Theta \cup \Omega$ is consistent and all its subsets are consistent, then we must consider some $X \subseteq \Theta \cup \Omega$ where $X \cup \{f_1\}$ is not consistent. Hence, $X \cup \{f_2\}$ must be consistent. As $\Theta \cup \Omega \in CandPgm_{AM}(\mathcal{I}_2)$, we know that $X \in \{X \setminus \{f_2\} \mid X \in CandPgm_{AM}(\mathcal{I}_2)\}$ iff $X \cup \{f_2\}$ is consistent, so it must be in that set. However, $X \notin \{X \setminus \{f_1\} \mid X \in CandPgm_{AM}(w, \mathcal{I}_1)\}$ as $X \cup \{f_1\}$ is not consistent — this is a contradiction. □

We now define the class of AM-based Operators, denoted **AMO**. Essentially, this operator selects one of the candidate programs in a deterministic fashion.

Definition 17 (**AM-based Operators**) A belief revision operator $\bullet$ is an "AM-based" operator ($\bullet \in$ **AMO**) iff given program $\mathcal{I} = (\Pi_{EM}, \Pi_{AM}, af)$ and input (f, af'), the revision is defined as $\mathcal{I} \bullet (f, af') = (\Pi_{EM}, \Pi'_{AM}, af')$, where $\Pi'_{AM} \in CandPgm_{AM}(\mathcal{I} \cup (f, af'))$.

Finally, we are able to prove our representation theorem for AM-based belief revision. This theorem follows directly from the results presented in this section.

Theorem 1 (**AM Representation Theorem**) *An operator $\bullet$ belongs to class **AMO** iff it satisfies Inclusion, Vacuity, Consistency Preservation, Weak Success, Pertinence, and Uniformity 1.*

Proof (Sketch) (If) By the definition of **AMO**, Vacuity and Weak Success follow trivially. Further, Lemma 2 shows that Inclusion, Consistency Preservation and Pertinence are satisfied, while Lemma 3 shows that Uniformity 1 is satisfied.
(Only-If) Suppose BWOC that an operator $\bullet$ satisfies all postulates from the statement and $\bullet \notin$ **AFO**. Then, one of the following conditions must hold: (i) it does not satisfy Lemma 2; (ii) it does not satisfy Lemma 3. However, by those previous arguments, if it satisfies all postulates from the statement, these arguments must be true as well — hence a contradiction. □

Example 16 Recall the AM knowledge base of the Fig. 1. We want to add $\theta_{3a} = l$ and $\theta_{3b} = \neg l$ to AM. Also recall $\mathcal{K}_{EM}$ defined in Example 2. Let $af(\theta_{3a}) = a$ and $af(\theta_{3a}) = b$. The input is $(f, af') = (\{\theta_{3a}, \theta_{3b}\}, af')$ where af' is the new annotation function. The program $\mathcal{I} \cup (f, af') = (\Pi_{EM}, \Pi_{AM} \cup \{f\}, af')$ will be inconsistent because of f_8. The AM-based belief revision $\mathcal{I} \bullet (f, af')$ will remove either θ_{3a} or θ_{3b}. The resulting program $\mathcal{I}'$ will be consistent.

We add $\theta_{3a} = l$ and $\theta_{3b} = \neg l$ to the AM knowledge base of the Fig. 1 and $f_8 = a \wedge b : 0.4 \pm 0.1$ to the $\mathcal{K}_{EM}$ defined in the Example 2. Also, let $af(\theta_{3a}) = a$ and $af(\theta_{3a}) = b$. In this scenario, the AM-based revision operator will remove either θ_{3a} or θ_{3b}. The resulting knowledge base will be consistent.

4.4 Annotation function-based belief revision

In this section we attack the belief revision problem from a different angle: adjusting the annotation function. The advantage to changing the annotation function is that we might

not need to discard an entire fact or strict rule from the argumentation model. Consider the following example.

Example 17 Let us consider two contradictory facts in an AM: a and $\neg a$ such that $af(a) = q \wedge r$ and $af(\neg a) = r \wedge s$. If we assume that q, r, s are the only atoms in the EM, then we know that a occurs under the environmental worlds $\{q, r\}$ and $\{q, r, s\}$, and that $\neg a$ occurs under the environmental worlds $\{r, s\}$ $\{q, r, s\}$.

Clearly, they cannot both be true in world $\{q, r, s\}$. Hence, a new annotation formula af' where $af'(a) = q \wedge r$ and $af'(\neg a) = r \wedge s \wedge \neg for(\{q, r, s\})$ easily solves the conflict (note that $for(w)$ specifies a formula satisfied by exactly world w). Note that we did not have to remove $\neg a$ from the knowledge base, which means that this information is not completely lost. In other word, the main difference between the AM-based belief revision and adjusting the Annotation function is that the later model allows more delicate changes to be made in order to preserve the information gathered in AM.

We also note that modifications of the annotation function can be viewed as a generalization of AM modification. Consider the following:

Example 18 Consider again the present facts a and $\neg a$ in the AM. Assuming that this causes an inconsistency (that is, there is at least one world in which they both hold), one way to resolve it would be to remove one of these two literals. Suppose $\neg a$ is removed; this would be equivalent to setting $af(\neg a) = \bot$ (where $\bot$ represents a contradiction in the language of the EM).

In this section, we introduce a set of postulates for reasoning about annotation function-based belief revision. As in the previous section, we then go on to provide a class of operators that satisfy all the postulates and show that this class includes all operators satisfying the postulates.

As in this section we are only conducting annotation function revisions, for $\mathcal{I} = (\Pi_{EM}, \Pi_{AM}, af)$ and input (f, af') we denote the revision as follows: $\mathcal{I}\blacklozenge(f, af') = (\Pi_{EM}, \Pi'_{AM}, af'')$ where $\Pi'_{AM} = \Pi_{AM} \cup \{f\}$ and af'' is the revised annotation function. Further, in this section, we often refer to "removing elements of Π_{AM}" to refer to changes to the annotation function that cause certain elements of the Π_{AM} to not have their annotations satisfied in certain EM worlds. Further, as we are looking to change the annotation function for a specific subset of facts and strict rules, we specify these subsets with the following notation.

- $wld(f) = \{w \mid w \models f\}$ – the set of worlds that satisfy formula f; and
- $for(w) = \bigwedge_{a \in w} a \wedge \bigwedge_{a \notin w} \neg a$ – the formula that has w as its only model.
- $\Pi^{\mathcal{I}}_{AM}(w) = \{f \in \Theta \cup \Omega \mid w \models af(f)\}$

Intuitively, $\Pi^{\mathcal{I}}_{AM}(w)$ is the subset of facts and strict rules in Π_{AM} whose annotations are true in EM world w.

4.4.1 Postulates for revising the annotation function

Just as we did for AM-based belief revision, here we introduce rationality postulates for annotation function based belief revision. We note that except for vacuity, consistency

preservation, and weak success, the postulates are defined in a different manner from the AM postulates. *The key difference between the AM-based and the AF-based postulates is that AF postulates consider subsets of the AM that occur in certain the environmental conditions — as opposed to considering the entire analytical model as a whole.* In this way, the AF-based postulates will give rise to a more fine-grained revision of the overall knowlegebase than the more coarse-grain AM-based approach.

AF inclusion For $\mathcal{I}\blacklozenge(f, af') = (\Pi_{EM}, \Pi_{AM} \cup \{f\}, af'')$,
$\forall g \in \Pi_{AM}, wld\big(af''(g)\big) \subseteq wld(af'(g))$.
This postulate states that, for any element in the AM, the worlds that satisfy its annotation after the revision are a subset of the original set of worlds satisfying the annotation for that element.

AF vacuity If $\mathcal{I} \cup (f, af')$ is consistent, then $\mathcal{I}\blacklozenge(f, af') \subseteq \mathcal{I} \cup (f, af')$.
This is the same as for the AM version of the postulate: no change is made if the program is consistent with the added input.

AF consistency preservation If $\mathcal{I}$ is consistent, then $\mathcal{I}\blacklozenge(f, af')$ is also consistent.
Again, as with the AM version, the operator maintains a consistent program.

AF weak success If $\mathcal{I} \cup (f, af')$ is consistent, then $(f, af') \in \mathcal{I}\blacklozenge(f, af')$.
The input must be contained in the revised program if it does not cause inconsistencies.

For a given EM world, if a portion of the associated AM knowledge base is removed by the operator, then there exists a subset of the remaining knowledge base that is not consistent with the removed element and f.

AF pertinence For $\mathcal{I}\blacklozenge(f, af') = (\Pi_{EM}, \Pi_{AM} \cup \{f\}, af'')$, for each $w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af'))$, we have $X_w = \{h \in \Theta \cup \Omega \mid w \models af''(h)\}$; for each $g \in \Pi_{AM}(w) \setminus X_w$ there exists $Y_w \supseteq X_w \cup \{f\}$ s.t. Y_w is consistent and $Y_w \cup \{g\}$ is inconsistent.
For a given EM world, if a portion of the associated AM knowledge base is removed by the operator, then there exists a superset of the remaining knowledge base that is not consistent with the removed element and f.

AF uniformity 1 Let $(f, af'_1), (g, af'_2)$ be two inputs where
$\mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af'_1)) = \mathcal{W}^I_{EM}(\mathcal{I} \cup (g, af'_2))$; for all $w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af'))$ and for all $X \subseteq \Pi_{AM}(w)$; if $\{x \mid x \in X \cup \{f\}, w \models af'_1(x)\}$ is inconsistent iff $\{x \mid x \in X \cup \{g\}, w \models af'_2(x)\}$ is inconsistent, then for each $h \in \Pi_{AM}$, we have that:

$$\{w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af'_1)) \mid w \models af'_1(h) \wedge \neg af''_1(h)\} =$$

$$\{w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (g, af'_2)) \mid w \models af'_2(h) \wedge \neg af''_2(h)\}.$$

If two inputs result in the same set of EM worlds leading to inconsistencies in an AM knowledge base, and the consistency between analogous subsets (when joined with the respective input) are the same, then the models removed from the annotation of a given strict rule or fact are the same for both inputs.

AF uniformity 2 Let (f, af'_1), (g, af'_2) be two inputs where $\mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af'_1)) = \mathcal{W}^I_{EM}(\mathcal{I} \cup (g, af'_2))$; for all $w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af'))$and for all $X \subseteq \Pi_{AM}(w)$; if $\{x \mid x \in X \cup \{f\}, w \models af'_1(x)\}$ is inconsistent iff $\{x \mid x \in X \cup \{g\}, w \models af'_2(x)\}$ is inconsistent, then

$$\{w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af'_1)) \mid w \models af'_1(h) \wedge af''_1(h)\} =$$

$$\{w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (g, af'_2)) \mid w \models af'_2(h) \wedge af''_2(h)\}.$$

If two inputs result in the same set of EM worlds leading to inconsistencies in an AM knowledge base, and the consistency between analogous subsets (when joined with the respective input) are the same, then the models retained in the the annotation of a given strict rule or fact are the same for both inputs.

4.4.2 AF-based revision operator

In this section, we introduce a class of operators for revising a DeLP3E program. Unlike the AM revision, this fine-grained approach requires an adjustment of the conditions in which elements of Π_{AM} can hold true. Hence, any subset of Π_{AM} associated with a world in $\mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af'))$ must be modified by the operator in order to remain consistent. So, for such a world w, we introduce the annotation function version of the set of candidate replacement programs for $\Pi_{AM}(w)$ in order to maintain consistency and satisfy the Inclusion postulate.

$$\begin{aligned} CandPgm_{af}(w, \mathcal{I}) = \\ &\{\Pi'_{AM} \mid \Pi'_{AM} \subseteq \Pi_{AM}(w) \text{ s.t. } \Pi'_{AM} \text{ is consistent and} \\ &\nexists \Pi''_{AM} \subseteq \Pi_{AM}(w) \textit{ s.t. } \Pi''_{AM} \supset \Pi'_{AM} \text{ s.t. } \Pi''_{AM} \\ &\text{is consistent}\} \end{aligned}$$

Intuitively, for each world w, this is the set of is a maximal consistent subsets of $\Pi^{\mathcal{I}}_{AM}(w)$. However, unlike with AM based belief revision, the candidate replacement program are specified for specific worlds - **this in turn enables a more "surgical" adjustment to the overall knowledgebase than AM belief revision.** This is due to the fact that in AM revision, components of the analytical model are deemed to no longer hold in any world as opposed to a specific subset of worlds.

Before introducing our operator, we define some preliminary notation. Let $\Phi : \mathcal{W}_{EM} \rightarrow 2^{[\Theta] \cup [\Omega]}$. Recall that sets of all facts and rules are denoted by $[\Theta]$ and $[\Omega]$ respectively. For each formula h in $\Pi_{AM} \cup \{f\}$, where f is part of the input, we define:

$$newFor(h, \Phi, \mathcal{I}, (f, af')) = af'(h) \wedge \bigwedge_{w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af')) \mid h \notin \Phi(w)} \neg for(w)$$

Intuitively, *newFor* eliminate inconsistency by adding the negation of the formulas whose only models are the inconsistent words. These inconsistent words are the result of adding the input f to the existing program $\mathcal{I}$.

Now we define the class of operators called **AFO**. We show that membership in **AFO** is a necessary and sufficient condition for satisfying all postulates introduced in this paper. The supporting Lemmas and their associated proofs are included in the Appendix.

Definition 18 (**AF-based Operators**) A belief revision operator $\blacklozenge$ is an "annotation function-based" (or af-based) operator ($\blacklozenge \in$ **AFO**) iff given program $\mathcal{I} = (\Pi_{EM}, \Pi_{AM}, af)$

and input (f, af'), the revision is defined as $\mathcal{I}\blacklozenge(f, af') = (\Pi_{EM}, \Pi_{AM} \cup \{f\}, af'')$, where:

$$\forall h, af''(h) = newFor(h, \Phi, \mathcal{I}, (f, af'))$$

where $\forall w \in \mathcal{W}_{EM}, \Phi(w) \in CandPgm_{af}(w, \mathcal{I} \cup (f, af'))$.

Theorem 2 (**Annotation Function Representation Theorem**) *An operator* $\blacklozenge$ *belongs to class* ***AFO*** *iff it satisfies Inclusion, Vacuity, Consistency Preservation, Weak Success, Pertinence, and Uniformity 1.*

Proof (Sketch) (If) By the fact that formulas associated with worlds in the set $\mathcal{W}^{\mathcal{I}}_{EM}(\mathcal{I} \cup (f, af'))$ are considered in the change of the annotation function, Vacuity and Weak Success follow trivially. Further, Lemma 8 shows that Inclusion, Consistency Preservation, and Pertinence are satisfied while Lemma 9 shows that Uniformity 1 is satisfied.
(Only-If) Suppose BWOC that an operator $\blacklozenge$ satisfies all postulates and $\blacklozenge \notin$ **AFO**. Then, one of four conditions must hold: (i) it does not satisfy Lemma 8 or (ii) it does not satisfy Lemma 9. However, by those previous arguments, if it satisfies all postulates, these arguments must be true as well – hence a contradiction. □

5 Case study: an application in cyber security

In this section we develop a complete example of how the DeLP3E framework can be used to deal with a cyber-attribution problem. In this scenario, a cyber attack has been detected and we want to determine who is responsible for it.

5.1 Model for the attribution problem

To specify the model we need to specify the environmental model, the analytical model, and the annotation function. First we identify two special subsets of the set of constants ($\mathbf{C}$) for this application: $\mathbf{C}_{act}$ and $\mathbf{C}_{ops}$, which specify the actors that could conduct cyber-operations and the operations themselves, respectively:

$$\mathbf{C}_{act} = \{baja, krasnovia, mojave\}$$
$$\mathbf{C}_{ops} = \{worm123\}$$

That is, the possible actors are the states of *baja*, *krasnovia* and *mojave*, and the only operation that we consider they can conduct is a *worm123* attack.

Next, we need to specify the sets of predicates, $\mathbf{P}_{EM}$, the predicates for the environmental model, and $\mathbf{P}_{AM}$, the predicates for the analytical model. These are given in Fig. 5, which presents all the predicates with variables. The following are examples of ground atoms over those predicates; again, we distinguish between the subset of ground atoms from the environmental model $\mathbf{G}_{EM}$ and the ground atoms from the analytical model $\mathbf{G}_{AM}$:

$$\mathbf{G}_{EM}: \; origIP(mw123sam1, krasnovia), mwHint(mw123sam1, krasnovia), inLgConf(krasnovia, baja), mseTT(krasnovia, 2)$$
$$\mathbf{G}_{AM}: \; evidOf(mojave, worm123), motiv(baja, krasnovia), expCw(baja), tgt(krasnovia, worm123)$$

$\mathbf{P}_{EM}$:	$origIP(M, X)$	Malware M originated from an IP address belonging to actor X.
	$malwInOp(M, O)$	Malware M was used in cyber-operation O.
	$mwHint(M, X)$	Malware M contained a hint that it was created by actor X.
	$compilLang(M, C)$	Malware M was compiled in a system that used language C.
	$nativLang(X, C)$	Language C is the native language of actor X.
	$inLgConf(X, X')$	Actors X, X' are in a larger conflict with each other.
	$mseTT(X, N)$	The number of top-tier math-science-engineering universities in country X is at least N.
	$infGovSys(X, M)$	Systems belonging to actor X were infected with malware M.
	$cybCapAge(X, N)$	Actor X has had a cyber-warfare capability for N years or less.
	$govCybLab(X)$	Actor X has a government cyber-security lab.
$\mathbf{P}_{AM}$:	$condOp(X, O)$	Actor X conducted cyber-operation O.
	$evidOf(X, O)$	There is evidence that actor X conducted cyber-operation O.
	$motiv(X, X')$	Actor X had a motive to launch a cyber-attack against actor X'.
	$isCap(X, O)$	Actor X is capable of conducting cyber-operation O.
	$tgt(X, O)$	Actor X was the target of cyber-operation O.
	$hasMseInvest(X)$	Actor X has a significant investment in math-science-engineering education.
	$expCw(X)$	Actor X has experience in conducting cyber-operations.

Fig. 5 Predicate definitions for the environment and analytical models in the cyber attribution example

$\mathbf{P}_{AM}$ and the set of constants provides all the information we need for the analytical model. However, there is more to the environmental model than just $\mathbf{P}_{EM}$ and the constants. We need to specify the probabilities of formulae. This information is given by the following set of probabilistic formulae $\mathcal{K}_{EM}$:

$$
\begin{aligned}
f_1 &= \mathit{govCybLab}(\mathit{baja}) : 0.8 \pm 0.1 \\
f_2 &= \mathit{cybCapAge}(\mathit{baja}, 5) : 0.2 \pm 0.1 \\
f_3 &= \mathit{mseTT}(\mathit{baja}, 2) : 0.8 \pm 0.1 \\
f_4 &= \mathit{mwHint}(\mathit{mw123sam1}, \mathit{mojave}) \\
&\quad \wedge \mathit{compilLang}(\mathit{worm123}, \mathit{english}) : 0.7 \pm 0.2 \\
f_5 &= \mathit{malwInOp}(\mathit{mw123sam1}, \mathit{worm123}) \\
&\quad \wedge \mathit{malwareRel}(\mathit{mw123sam1}, \mathit{mw123sam2}) \\
&\quad \wedge \mathit{mwHint}(\mathit{mw123sam2}, \mathit{mojave}) : 0.6 \pm 0.1 \\
f_6 &= \mathit{inLgConf}(\mathit{baja}, \mathit{krasnovia}) \vee \neg \mathit{cooper}(\mathit{baja}, \mathit{krasnovia}) : 0.9 \pm 0.1 \\
f_7 &= \mathit{origIP}(\mathit{mw123sam1}, \mathit{baja}) : 1 \pm 0
\end{aligned}
$$

Given this probabilistic information, we can demonstrate the linear programming approach to the *maximum entailment* problem defined in Definition 2. Consider knowledge base $\mathcal{K}'_{EM}$ and a set of ground atoms restricted to those that appear in that program. Hence, we have the following worlds:

$$
\begin{aligned}
w_1 &= \{govCybLab(baja), cybCapAge(baja, 5), mseTT(baja, 2)\} \\
w_2 &= \{govCybLab(baja), cybCapAge(baja, 5)\} \\
w_3 &= \{govCybLab(baja), mseTT(baja, 2)\} \\
w_4 &= \{cybCapAge(baja, 5), mseTT(baja, 2)\} \\
w_5 &= \{cybCapAge(baja, 5)\} \\
w_6 &= \{govCybLab(baja)\} \\
w_7 &= \{mseTT(baja, 2)\} \\
w_8 &= \emptyset
\end{aligned}
$$

and suppose we wish to compute the probability for formula:

$$q = govCybLab(baja) \vee mseTT(baja, 2)$$

For each formula in $\mathcal{K}_{EM}$ we have a constraint, and for each world above we have a variable. An objective function is created based on the worlds that satisfy the query formula (in this case, worlds $w_1, w_2, w_3, w_4, w_6, w_7$). Hence, $\mathsf{EP\text{-}LP\text{-}MIN}(\mathcal{K}'_{EM}, q)$ can be written as follows:

$$
\begin{array}{rcl}
\max & x_1 + x_2 + x_3 + x_4 + x_6 + x_7 & w.r.t.: \\
0.7 \leq & x_1 + x_2 + x_3 + x_6 & \leq 0.9 \\
0.1 \leq & x_1 + x_2 + x_4 + x_5 & \leq 0.3 \\
0.8 \leq & x_1 + x_3 + x_4 + x_7 & \leq 1 \\
& x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 & = 1
\end{array}
$$

From this, we can solve $\mathsf{EP\text{-}LP\text{-}MAX}(\mathcal{K}'_{EM}, q)$ and, after an easy modification, $\mathsf{EP\text{-}LP\text{-}MIN}(\mathcal{K}'_{EM}, q)$, and obtain the solution 0.9 ± 0.1.

Now, given $\mathbf{P}_{AM}$ and $\mathbf{C}$, we can assemble the ground argumentation framework of Fig. 6 as a sample Π_{AM}. From this argumentation framework, we can build the following arguments:

$$
\begin{array}{rl}
\langle \mathcal{A}_1, condOp(baja, worm123) \rangle & \mathcal{A}_1 = \{\theta_{1a}, \delta_{1a}\} \\
\langle \mathcal{A}_2, condOp(baja, worm123) \rangle & \mathcal{A}_2 = \{\phi_1, \phi_2, \delta_4, \omega_{2a}, \theta_{1a}, \theta_2\} \\
\langle \mathcal{A}_3, condOp(baja, worm123) \rangle & \mathcal{A}_3 = \{\phi_1, \delta_2, \delta_4\} \\
\langle \mathcal{A}_4, condOp(baja, worm123) \rangle & \mathcal{A}_4 = \{\phi_2, \delta_3, \theta_2\} \\
\langle \mathcal{A}_5, isCap(baja, worm123) \rangle & \mathcal{A}_5 = \{\phi_1, \delta_4\} \\
\langle \mathcal{A}_6, \neg condOp(baja, worm123) \rangle & \mathcal{A}_6 = \{\delta_{1b}, \theta_{1b}, \omega_{1a}\} \\
\langle \mathcal{A}_7, \neg isCap(baja, worm123) \rangle & \mathcal{A}_7 = \{\phi_3, \delta_{5a}\}
\end{array}
$$

Note that:

$$\langle \mathcal{A}_5, isCap(baja, worm123) \rangle$$

is a sub-argument of both

$$\langle \mathcal{A}_2, condOp(baja, worm123) \rangle$$

$$
\begin{array}{lll}
\Theta: & \theta_{1a} = & evidOf(baja, worm123) \\
& \theta_{1b} = & evidOf(mojave, worm123) \\
& \theta_2 = & motiv(baja, krasnovia) \\
\hline
\Omega: & \omega_{1a} = & \neg condOp(baja, worm123) \leftarrow condOp(mojave, worm123) \\
& \omega_{1b} = & \neg condOp(mojave, worm123) \leftarrow condOp(baja, worm123) \\
& \omega_{2a} = & condOp(baja, worm123) \leftarrow \\
& & \quad evidOf(baja, worm123), \\
& & \quad isCap(baja, worm123), \\
& & \quad motiv(baja, krasnovia), \\
& & \quad tgt(krasnovia, worm123) \\
& \omega_{2b} = & condOp(mojave, worm123) \leftarrow \\
& & \quad evidOf(mojave, worm123), \\
& & \quad isCap(mojave, worm123), \\
& & \quad motiv(mojave, krasnovia), \\
& & \quad tgt(krasnovia, worm123) \\
\hline
\Phi: & \phi_1 = & hasMseInvest(baja) \prec \\
& \phi_2 = & tgt(krasnovia, worm123) \prec \\
& \phi_3 = & \neg expCw(baja) \prec \\
\hline
\Delta: & \delta_{1a} = & condOp(baja, worm123) \prec evidOf(baja, worm123) \\
& \delta_{1b} = & condOp(mojave, worm123) \prec evidOf(mojave, worm123) \\
& \delta_2 = & condOp(baja, worm123) \prec isCap(baja, worm123) \\
& \delta_3 = & condOp(baja, worm123) \prec \\
& & \quad motiv(baja, krasnovia), \\
& & \quad tgt(krasnovia, worm123) \\
& \delta_4 = & isCap(baja, worm123) \prec hasMseInvest(baja) \\
& \delta_{5a} = & \neg isCap(baja, worm123) \prec \neg expCw(baja) \\
& \delta_{5b} = & \neg isCap(mojave, worm123) \prec \neg expCw(mojave)
\end{array}
$$

Fig. 6 A ground argumentation framework

and

$$\langle \mathcal{A}_3, \mathit{condOp}(\mathit{baja}, \mathit{worm123}) \rangle$$

The following are some of the attack relationships between these arguments: $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$, and $\mathcal{A}_4$ all attack $\mathcal{A}_6$; $\mathcal{A}_5$ attacks $\mathcal{A}_7$; and $\mathcal{A}_7$ attacks $\mathcal{A}_2$.

In Fig. 7 we show an another example of a knowledge base for the attribution problem, this time with a non-ground argumentation system.

With the environmental and analytical models specified, the remaining component of the model is the annotation function; one suitable annotation function is given in Fig. 8. Consider worlds $w_1, \ldots, w_8$ along with the argument $\langle \mathcal{A}_5, \mathit{isCap}(\mathit{baja}, \mathit{worm123}) \rangle$. This argument is valid in worlds w_1, w_2, w_3, w_4, w_6, and w_7. Similarly, worlds w_3, w_6, and w_7 are warranting scenarios for argument $\langle \mathcal{A}_5, \mathit{isCap}(\mathit{baja}, \mathit{worm123}) \rangle$ and

$$\mathit{nec}(\mathit{isCap}(\mathit{baja}, \mathit{worm123})) = \{w_3, w_6, w_7\}$$

while

$$\mathit{poss}(\mathit{isCap}(\mathit{baja}, \mathit{worm123})) = \{w_1, w_2, w_3, w_4, w_6, w_7\}$$

5.2 Applying entailment to the cyber-attribution problem

We now discuss how finding tight bounds on the entailment probability can be applied to the cyber-attribution problem. Following the domain-specific notation introduced in

$$
\begin{array}{lll}
\Theta: & \theta_1 = & evidOf(baja, worm123) \\
& \theta_2 = & motiv(baja, krasnovia) \\
\hline
\Omega: & \omega_1 = & \neg condOp(X, O) \leftarrow condOp(X', O), X \neq X' \\
& \omega_2 = & condOp(X, O) \leftarrow evidOf(X, O), isCap(X, O), \\
& & \quad motiv(X, X'), tgt(X', O), X \neq X' \\
\hline
\Phi: & \phi_1 = & hasMseInvest(baja) \prec \\
& \phi_2 = & tgt(krasnovia, worm123) \prec \\
& \phi_3 = & \neg expCw(baja) \prec \\
\hline
\Delta: & \delta_1 = & condOp(X, O) \prec evidOf(X, O) \\
& \delta_2 = & condOp(X, O) \prec isCap(X, O) \\
& \delta_3 = & condOp(X, O) \prec motiv(X, X'), tgt(X', O) \\
& \delta_4 = & isCap(X, O) \prec hasMseInvest(X) \\
& \delta_5 = & \neg isCap(X, O) \prec \neg expCw(X)
\end{array}
$$

Fig. 7 A non-ground argumentation framework

the beginning of this case study (where the set of constants $\mathbf{C}$ includes two subsets: $\mathbf{C}_{act}$ and $\mathbf{C}_{ops}$, that specify the actors that could conduct cyber-operations and the operations themselves, respectively), we define a special case of the entailment problem as follows.

Definition 19 Let $\mathcal{I} = (\Pi_{EM}, \Pi_{AM}, af)$ be a DeLP3E program, $\mathcal{S} \subseteq \mathbf{C}_{act}$ (the set of "suspects"), $\mathcal{O} \in \mathbf{C}_{ops}$ (the "operation"), $\mathcal{E} \subseteq \mathbf{G}_{EM}$ (the "evidence"), and $\mathcal{D} \subseteq \mathbf{G}_{EM}$ (the "probabilistic fact").

An actor $\mathsf{A} \in \mathcal{S}$ is said to be a *most probable suspect* iff there does not exist $\mathsf{A}' \in \mathcal{S}$ such that $\mathbf{P}_{condOp(\mathsf{A}', \mathcal{O}), \mathcal{I}'} > \mathbf{P}_{condOp(\mathsf{A}, \mathcal{O}), \mathcal{I}'}$ where $\mathcal{I}' = (\Pi_{EM} \cup \Pi_{\mathcal{E}} \cup \Pi_{\mathcal{D}}, \Pi_{AM}, af')$ with $\Pi_{\mathcal{E}} = \bigcup_{c \in \mathcal{E}} \{c : 1 \pm 0\}$ and $\Pi_{\mathcal{D}} = \bigcup_{c \in \mathcal{D}} \{c : p \pm \epsilon\}$.

Note that $\mathbf{P}_{condOp(\mathsf{A}', \mathcal{O}), \mathcal{I}'}$ and $\mathbf{P}_{condOp(\mathsf{A}, \mathcal{O}), \mathcal{I}'}$ are midpoint of intervals $\mathbf{P}_{condOp(\mathsf{A}', \mathcal{O}), \mathcal{I}'} \pm \epsilon$ and $\mathbf{P}_{condOp(\mathsf{A}, \mathcal{O}), \mathcal{I}'} \pm \epsilon$. Alternative formulations are possible based on upper or lower bound of interval.

Given the above definition, we refer to $Q = (\mathcal{I}, \mathcal{S}, \mathcal{O}, \mathcal{E})$ as an *attribution query*, and A as an *answer* to Q. We note that in the above definition, the items of evidence are added

$$
\begin{array}{ll}
af(\theta_1) = & origIP(worm123, baja) \vee \big(malwInOp(worm123, o) \wedge \\
& \big(mwHint(worm123, baja) \vee (compilLang(worm123, c) \wedge nativLang(baja, c))\big)\big) \\
af(\theta_2) = & inLgConf(baja, krasnovia) \\
af(\omega_1) = & \mathsf{True} \\
af(\omega_2) = & \mathsf{True} \\
af(\phi_1) = & mseTT(baja, 2) \vee govCybLab(baja) \\
af(\phi_2) = & malwInOp(worm123, o') \wedge infGovSys(krasnovia, worm123) \\
af(\phi_3) = & cybCapAge(baja, 5) \\
af(\delta_1) = & \mathsf{True} \\
af(\delta_2) = & \mathsf{True} \\
af(\delta_3) = & \mathsf{True} \\
af(\delta_4) = & \mathsf{True} \\
af(\delta_5) = & \mathsf{True}
\end{array}
$$

Fig. 8 Example annotation function

to the environmental model with a probability of 1. While, in general, this may be the case, there are often instances in analysis of a cyber-operation where the evidence may be true with some degree of uncertainty. For this reason we allow for probabilistic facts in the definition.

To understand how uncertain evidence can be present in a cyber-security scenario, consider the following scenario:

> In Symantec's initial analysis of the Stuxnet worm, analysts found the routine designed to attack the S7-417 logic controller was incomplete, and hence would not function [13]. However, industrial control system expert Ralph Langner claimed that the incomplete code would run provided a missing data block is generated, which he thought was possible [27]. In this case, though the code was incomplete, uncertainty was clearly present regarding its usability.[5]

This situation provides a real-world example of the need to compare arguments — in this case, in the worlds where both arguments are valid, Langner's argument would likely defeat Symantec's by generalized specificity (the outcome, of course, will depend on the exact formalization of the two).

In Fig. 9 we give a simple, straightforward algorithm for attribution queries. The correctness of this algorithm clearly follows from the definitions above. We note that a key source of computational complexity lies in step 2, where all arguments supporting the hypothesis that each actor conducted the operation are computed *for each world in the EM*; this leads to a factor of $2^{|\mathbf{G}_{EM}|}$ (exponential in the number of ground atoms in the environmental model). However, we also note that this is equal to the time complexity required to write out a linear program for answering the entailment query.

Note that the exact approaches presented thus far for answering attribution queries experience exponential running times in the worst case. Hence, for the creation of a real-world system, we consider several practical approaches that can be taken to answer attribution queries $Q = (\mathcal{I}, \mathcal{S}, \mathcal{O}, \mathcal{E})$. We are currently exploring several of these ideas as we work to build a system for cyber-attribution based on DeLP3E:

1. *Approximating the warranting formula:* Instead of inspecting all possible classical dialectical trees as in Approach 1, either a subset of trees can be computed according to a given heuristic or an anytime approach can be adopted to select such a subset $\mathcal{F}'$. The computations with respect to F' will then yield sound approximations relative to the full forest $\mathcal{F}$, which means that all probability intervals will be supersets of the exact intervals.
2. *Approximating the probability:* Another alternative to Approach 1 is to apply approximation algorithms to the formula; for instance:
   (a) Approximate satisfiability: if the formula is unsatisfiable, then the warranting probability is zero;
   (b) A lower bound on the warranting probability can be obtained from a subset of possible worlds (k most probable worlds, random sample of worlds, etc.).
3. *"What-if" Reasoning:* Given a set $\mathcal{W}_{int}$ of worlds of interest and a warranting formula ϕ (computed using any of the above approaches), each world can be checked to

[5] Langner was later vindicated by the discovery of an older sample, Stuxnet 0.5, which generated the data block [5].

Algorithm SFWD_ATTRIB$(\mathcal{I}, \mathcal{S}, \mathcal{O}, \mathcal{E})$
1. Let $\mathcal{I}' = (\Pi_{EM}, \Pi_{AM} \cup \mathcal{E}, af')$, where $af'(c) = \top$ if $c \in \mathcal{E}$ and $af'(c) = af(c)$ otherwise.
2. For each $\mathsf{A} \in \mathcal{S}$ and $w \in \mathcal{W}_{EM}$ do
3. $\quad Pos := \mathcal{F}^*_w(condOp(\mathsf{A}, \mathcal{O}))$ w.r.t. $\mathcal{I}'$;
4. $\quad Neg := \mathcal{F}^*_w(\neg condOp(\mathsf{A}, \mathcal{O}))$ w.r.t. $\mathcal{I}'$;
5. For each $\mathsf{A} \in \mathcal{S}$ do
6. $\quad m := nec(condOp(\mathsf{A}, \mathcal{O}))$ w.r.t. Π_{EM} and af' (computed using Pos);
7. $\quad c := poss(condOp(\mathsf{A}, \mathcal{O}))$ w.r.t. Π_{EM} and af' (computed using Neg);
8. $\quad P_{\mathsf{A}} := \mathbf{P}_{condOp(\mathsf{A},\mathcal{O}),\mathcal{I}'}$;
9. Return $\arg\max_{\mathsf{A}}(P_{\mathsf{A}})$.

Fig. 9 A straightforward algorithm for finding a solution to an attribution query

see which literals $condOp(A_i, \mathcal{O})$, with $A_i \in \mathcal{S}$, are warranted. That is, instead of computing probability of attribution, the attribution literal is analyzed in each world of interest.

6 Conclusions

In this paper we introduced the DeLP3E framework, consisting of an environmental model, an analytical model, and an annotation function that relates the two. DeLP3E is an extension of the PreDeLP language in which sentences can be be annotated with probabilistic events. Such events are connected to a probabilistic model, allowing a clear separation of interests between certain and uncertain knowledge while allowing uncertainty to be captured and incorporated into reasoning. After presenting the language, we focused on characterizing belief revision operations over DeLP3E knowledge bases. We presented two sets of postulates, both inspired by the postulates that were developed for non-prioritized revision of classical belief bases. The first set of postulates provides a coarse approach that assumes that revision operations only allow changes to the analytical model, while the second is a finer-grained approach based on modifications to the annotation function. We then proceeded to study constructions of operators based on these postulates, and prove that they are equivalent to their characterizations by the respective postulates.

This paper makes a number of contributions to the literature of both argumentation and belief revision. First, this paper contains the most complete description of DeLP3E yet published.[6] This is a contribution to the study of probabilistic argumentation, and one that, with the separation between the argumentation system and the probabilistic information in the environmental model, makes it unique. Second, this paper presents two approaches to belief revision in DeLP3E. This is a contribution to the study of the relationship between argumentation systems and belief revision, one that views the problem from the position of structured argumentation. While the study of revision of the annotation function is specific to DeLP3E, the study of the revision of the analytical model will be relevant to all argumentation systems that combine strict and defeasible elements, such as DeLP [15], PreDeLP [30] and ASPIC+ [31, 33]. Finally, the paper presents an extended case study of the application of DeLP3E to the attribution problem. This is a contribution to both the argumentation

[6] DeLP3E was briefly introduced in [41] and [40] as a solution to the attribution problem.

literature, in showing how argumentation can be applied to a complex real-world problem, and to the cyber security literature, suggesting tools that can be used to address this problem. As part of the case study we considered a special kind of query, called an attribution query, that is useful in tackling the problem of attributing responsibility to entities given a cyber event of interest. This is a further contribution to the cyber security literature.

After this initial proposal, there remains much work to be done with DeLP3E. As future work, we plan to study other kinds of belief revision operators, including more general ones that allow the modification of the environmental model along with the other two components, as well as revision operators that function at different levels of granularity. Furthermore, we are in the final stages of producing an implementation of the system — important future work involves focusing on scalable inference algorithms and testing them on real-world data from the cyber security domain. The last thing to note is that, as discussed above, DeLP3E is less a specific formal system, and more a family of systems in which the analytical model and the environmental model are instantiated in different ways. Here we chose to use Nilsson's probabilistic logic to capture the world in the environmental model, but it is possible to use other frameworks for this purpose; for instance, Markov Logic networks [35] would be an interesting choice. Similarly, here we chose to use PreDeLP to build the analytical model. We can easily envisage versions of DeLP3E that use frameworks other than PreDeLP for the analytical model. For instance, an abstract argumentation model [3], an argumentation model that includes uncertain consideration in defeat relationships (such as a probabilistic argumentation model [28] or possibilistic argumentation model [4]) and we might also associate varying notions of strength with attack relations as in [8]. All of these are potential routes for future work.

Acknowledgments This work was supported by UK EPSRC grant EP/J008346/1—"PrOQAW", by ERC grant 246858—"DIADEM", funds provided by CONICET and Universidad Nacional del Sur, Argentina, by NSF grant #1117761, by the Army Research Office under the Science of Security Lablet grant (SoSL) and project 2GDATXR042, and DARPA project R.0004972.001. The opinions in this paper are those of the authors and do not necessarily reflect the opinions of the funders, the U.S. Military Academy, or the U.S. Army.

Appendix

In this appendix we provide some complementary material that was not included in the main body of the paper to enhance readability. Specifically, the results of this appendix support the proof of Theorem 2 in Section 4.4 (the representation theorem for AF-based belief revision).

First, we give the annotation function revision versions of Proposition 4.

Proposition 5 *For operator* $\blacklozenge$ *such that* $\mathcal{I}\blacklozenge(f, af') = (\Pi_{EM}, \Pi_{AM} \cup \{f\}, af'')$ *and* $\forall w$, *we have that* $\Pi_{AM}^{\mathcal{I}\blacklozenge(f,af')}(w) \subseteq \Pi_{AM}^{\mathcal{I}\cup(f,af')}(w)$, *it holds that* $\blacklozenge$ *satisfies AF Uniformity 1 iff it also satisfies AF Uniformity 2.*

Proof (If) Suppose BWOC that $\blacklozenge$ satisfies AF Uniformity 1 and does not satisfy AF Uniformity 2. Then $\forall w \in \mathcal{W}_{EM}^{I}(\mathcal{I} \cup (f, af'_1)) = \mathcal{W}_{EM}^{I}(\mathcal{I} \cup (g, af'_2))$ and $h \in \Pi_{AM}$ we have:

$$\{w \in \mathcal{W}_{EM}^{I}(\mathcal{I} \cup (f, af'_1)) \mid w \models af'_1(h)\} \cap \{w \in \mathcal{W}_{EM}^{I}(\mathcal{I} \cup (f, af'_1)) \mid w \models \neg af''_1(h)\} =$$
$$\{w \in \mathcal{W}_{EM}^{I}(\mathcal{I} \cup (g, af'_2)) \mid w \models af'_2(h)\} \cap \{w \in \mathcal{W}_{EM}^{I}(\mathcal{I} \cup (g, af'_2)) \mid w \models \neg af''_2(h)\}$$

and

$$\{w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af'_1)) \mid w \models af'_1(h)\} \cap \{w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af'_1)) \mid w \models af''_1(h)\} \neq$$
$$\{w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (g, af'_2)) \mid w \models af'_2(h)\} \cap \{w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (g, af'_2)) \mid w \models af''_2(h)\}$$

However, we note that $\forall h \in \Pi_{AM}$ we have $af'_1(h) = af'_2(h)$ and by the statement of the postulate $\mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af'_1)) = \mathcal{W}^I_{EM}(\mathcal{I} \cup (g, af'_2))$, we have the following:

$$\{w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af'_1)) \mid w \models \neg af''_1(h)\} =$$
$$\{w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (g, af'_2)) \mid w \models \neg af''_2(h)\}$$

Which implies a contradiction.
(Only-If) Mirrors the above claim. □

We now focus on complementary material relating to Section 4.4 on annotation function-based belief revision.

Lemma 4 *Given program $\mathcal{I}$ and input (f, af'), operator $\blacklozenge$ satisfies Inclusion and Consistency Preservation iff for $\mathcal{I}\blacklozenge(f, af') = (\Pi_{EM}, \Pi_{AM}, af'')$, for all $w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af'))$, there exists an element $X \in Cand Pgm_{af}(w, \mathcal{I} \cup (f, af'))$ s.t. $\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\} \subseteq X$.*

Proof (If) Suppose, BWOC, that there exists an element $X \in Cand Pgm_{af}(w, \mathcal{I} \cup (f, af'))$ s.t. $\{h \in \Theta \cup \Omega \cup \{f\} | w \models af''(h)\} \subseteq X$ but either Inclusion or Consistency Preservation is not satisfied. However, the elements of $Cand Pgm_{af}(w, \mathcal{I} \cup (f, af'))$ are all classically consistent and all subsets of $\Pi_{AM_{\mathcal{I} \cup (f, af')}}(w)$, which is a contradiction. (Only-If) Suppose, BWOC, that the operator satisfies both Inclusion and Consistency Preservation and there does not exist $X \in Cand Pgm_{af}(w, \mathcal{I} \cup (f, af'))$ s.t. $\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\} \subseteq X$. Then, $\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\}$ is a subset of $\Pi_{AM_{\mathcal{I} \cup (f, af')}}(w)$ and $\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\}$ is classically consistent. However, by definition, this would mean that it must also be a subset of an element in $Cand Pgm_{af}(w, \mathcal{I} \cup (f, af'))$. □

We now investigate the role that the set $Cand Pgm_{af}$ plays in showing the necessary and sufficient requirement for satisfying Pertinence.

Lemma 5 *Given program $\mathcal{I}$ and input (f, af'), operator $\blacklozenge$ satisfies Inclusion, Consistency Preservation, and Pertinence iff for $\mathcal{I}\blacklozenge(f, af') = (\Pi_{EM}, \Pi_{AM}, af'')$, for all $w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af'))$ we have $\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\} \in Cand Pgm_{af}(w, \mathcal{I} \cup (f, af'))$.*

Proof (If) Suppose, BWOC, that

$$\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\} \in Cand Pgm_{af}(w, \mathcal{I} \cup (f, af'))$$

(which, by Lemma 7, satisfies both Consistency and Inclusion) but does not satisfy Pertinence. As $|\Pi_{AM}(w) \setminus X_w| > 0$, then $f \in \{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\}$. This means that $\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\} \supseteq X_w \cup \{f\}$, which also yields a contradiction.
(Only-If) Suppose, BWOC, that the operator satisfies Inclusion, Consistency Preservation, and Pertinence but there exists w s.t. $\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\} \notin Cand Pgm_{af}(w, \mathcal{I} \cup (f, af'))$. By Lemma 7, $\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\} \in V = \{X \mid \exists Y \in Cand Pgm_{af}(w, \mathcal{I} \cup (f, af')) \wedge X \supseteq Y\}$. Hence, this would mean that

$\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\} \in Z = \{X \mid \exists Y \in CandPgm_{af}(w, \mathcal{I} \cup (f, af')) \wedge X \supset Y\}$ in this case. However, this would violate Lemma 7 — a contradiction. □

Lemma 6 *Let $\mathcal{I} = (\Pi_{EM}, \Pi_{AM}, af)$ be a consistent program, (f_1, af'_1), (f_2, af'_2) be two inputs, and $\mathcal{I}_i = (\Pi_{EM}, \Pi_{AM} \cup \{f_i\}, af'_i)$. If $\mathcal{W}^I_{EM}(\mathcal{I}_1) = \mathcal{W}^I_{EM}(\mathcal{I}_2)$, then for all $w \in \mathcal{W}^I_{EM}(\mathcal{I}_1)$ and all $X \subseteq \Pi_{AM}(w)$ we have that:*

1. *If $\{x \mid x \in X \cup \{f_1\}, w \models af'_1(x)\}$ is inconsistent $\Leftrightarrow \{x \mid x \in X \cup \{f_2\}, w \models af'_2(x)\}$ is inconsistent, then $\{X \setminus \{f_1\} \mid X \in CandPgm_{af}(w, \mathcal{I}_1)\} = \{X \setminus \{f_2\} \mid X \in CandPgm_{af}(w, \mathcal{I}_2)\}$.*
2. *If $\{X \setminus \{f_1\} \mid X \in CandPgm_{af}(w, \mathcal{I}_1)\} = \{X \setminus \{f_2\} \mid X \in CandPgm_{af}(w, \mathcal{I}_2)\}$ then $\{x \mid x \in X \cup \{f_1\}, w \models af'_1(x)\}$ is inconsistent $\Leftrightarrow \{x \mid x \in X \cup \{f_2\}, w \models af'_2(x)\}$ is inconsistent.*

Proof (If) Suppose BWOC that for all $w \in \mathcal{W}^I_{EM}(\mathcal{I}_1)$ and all $X \subseteq \Pi_{AM}(w)$; if $\{x \mid x \in X \cup \{f_1\}, w \models af'_1(x)\}$ is inconsistent iff $\{x \mid x \in X \cup \{f_2\}, w \models af'_2(x)\}$ is inconsistent, but there exists w s.t.:

$$\{X \setminus \{f_1\} \mid X \in CandPgm_{af}(w, \mathcal{I}_1)\} \neq \{X \setminus \{f_2\} \mid X \in CandPgm_{af}(w, \mathcal{I}_2)\}.$$

However, the pre-condition of this statement implies that $\{X \setminus \{f_1\} \mid X \subseteq CandPgm_{af}(w, \mathcal{I}_1)\} = \{X \setminus \{f_2\} \mid X \subseteq CandPgm_{af}(w, \mathcal{I}_2)$ which gives us a contradiction.
(Only-If) Suppose BWOC that for all w, $\{X \setminus \{f_1\} \mid X \in CandPgm_{af}(w, \mathcal{I}_1)\} = \{X \setminus \{f_2\} \mid X \in CandPgm_{af}(w, \mathcal{I}_2)\}$, but there exists a world $w \in \mathcal{W}^I_{EM}(\mathcal{I}_1)$ and set $X \subseteq \Pi_{AM}(w)$ s.t. exactly one of $\{x \mid x \in X \cup \{f_1\}, w \models af'_1(x)\}$, $\{x \mid x \in X \cup \{f_2\}, w \models af'_2(x)\}$ is inconsistent. As a first case, let us assume that $\Pi_{AM}(w) \cup \{f_1\}$ is consistent. As $\{X \setminus \{f_1\} \mid X \in CandPgm_{af}(w, \mathcal{I}_1)\} = \{X \setminus \{f_2\} \mid X \in CandPgm_{af}(w, \mathcal{I}_2)\}$, this implies that $\Pi_{AM}(w) \cup \{f_2\}$ must also be consistent as each of those sets must then have exactly one element. In this case a contradiction arises, hence both $\Pi_{AM}(w) \cup \{f_1\}$, $\Pi_{AM}(w) \cup \{f_2\}$ must be classically inconsistent. Now let us consider the other case. As $\Pi_{AM}(w)$ is consistent and all its subsets are consistent, then we must consider some $X \subseteq \Pi_{AM}(w)$ where $X \cup \{f_1\}$ is not consistent. Hence, $X \cup \{f_2\}$ must be consistent. As $\Pi_{AM}(w) \in CandPgm_{af}(w, \mathcal{I}_2)$, we know that $X \in \{X \setminus \{f_2\} \mid X \in CandPgm_{af}(w, \mathcal{I}_2)\}$ iff $X \cup \{f_2\}$ is consistent, so it must be in that set. However, $X \notin \{X \setminus \{f_1\} \mid X \in CandPgm_{af}(w, \mathcal{I}_1)\}$ as $X \cup \{f_1\}$ is not consistent — this is a contradiction. □

Lemma 7 *Given program $\mathcal{I}$ and input (f, af'), operator $\blacklozenge$ satisfies Inclusion and Consistency Preservation iff for $\mathcal{I}\blacklozenge(f, af') = (\Pi_{EM}, \Pi_{AM}, af'')$, for all $w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af'))$, there exists an element $X \in CandPgm_{af}(w, \mathcal{I} \cup (f, af'))$ s.t. $\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\} \subseteq X$.*

Proof (If) Suppose, BWOC, that there exists $X \in CandPgm_{af}(w, \mathcal{I} \cup (f, af'))$ s.t. $\{h \in \Theta \cup \Omega \cup \{f\} | w \models af''(h)\} \subseteq X$ but either Inclusion or Consistency Preservation is not satisfied. However, the elements of $CandPgm_{af}(w, \mathcal{I} \cup (f, af'))$ are all classically consistent and all subsets of $\Pi_{AM_{\mathcal{I} \cup (f, af')}}(w)$, which is a contradiction.
(Only-If) Suppose, BWOC, that the operator satisfies both Inclusion and Consistency Preservation and there does not exist $X \in CandPgm_{af}(w, \mathcal{I} \cup (f, af'))$ s.t. $\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\} \subseteq X$. Then, $\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\}$ is a subset of $\Pi_{AM_{\mathcal{I} \cup (f, af')}}(w)$ and $\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\}$ is classically consistent.

However, by definition, this would mean that it must also be a subset of an element in $Cand Pgm_{af}(w, \mathcal{I} \cup (f, af'))$. □

We now investigate the role that the set $Cand Pgm_{af}$ plays in showing the necessary and sufficient requirement for satisfying Pertinence.

Lemma 8 *Given program $\mathcal{I}$ and input (f, af'), operator $\blacklozenge$ satisfies Inclusion, Consistency Preservation, and Pertinence iff for $\mathcal{I}\blacklozenge(f, af') = (\Pi_{EM}, \Pi_{AM}, af'')$, for all $w \in \mathcal{W}^I_{EM}(\mathcal{I} \cup (f, af'))$ we have $\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\} \in Cand Pgm_{af}(w, \mathcal{I} \cup (f, af'))$.*

Proof (If) Suppose, BWOC, that

$$\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\} \in Cand Pgm_{af}(w, \mathcal{I} \cup (f, af'))$$

(which, by Lemma 7, satisfies both Consistency and Inclusion) but does not satisfy Pertinence. As $|\Pi_{AM}(w) \setminus X_w| > 0$, then $f \in \{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\}$. This means that $\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\} \supseteq X_w \cup \{f\}$, which also yields a contradiction.
(Only-If) Suppose, BWOC, that the operator satisfies Inclusion, Consistency Preservation, and Pertinence but there exists w s.t. $\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\} \notin Cand Pgm_{af}(w, \mathcal{I} \cup (f, af'))$. By Lemma 7, $\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\} \in V = \{X \mid \exists Y \in Cand Pgm_{af}(w, \mathcal{I} \cup (f, af')) \wedge X \supseteq Y\}$. Hence, this would mean that $\{h \in \Theta \cup \Omega \cup \{f\} \mid w \models af''(h)\} \in Z = \{X \mid \exists Y \in Cand Pgm_{af}(w, \mathcal{I} \cup (f, af')) \wedge X \supset Y\}$ in this case. However, this would violate Lemma 7 — a contradiction. □

Lemma 9 *Let $\mathcal{I} = (\Pi_{EM}, \Pi_{AM}, af)$ be a consistent program, (f_1, af'_1), (f_2, af'_2) be two inputs, and $\mathcal{I}_i = (\Pi_{EM}, \Pi_{AM} \cup \{f_i\}, af'_i)$. If $\mathcal{W}^I_{EM}(\mathcal{I}_1) = \mathcal{W}^I_{EM}(\mathcal{I}_2)$, then for all $w \in \mathcal{W}^I_{EM}(\mathcal{I}_1)$ and all $X \subseteq \Pi_{AM}(w)$ we have that:*

1. *If $\{x \mid x \in X \cup \{f_1\}, w \models af'_1(x)\}$ is inconsistent $\Leftrightarrow \{x \mid x \in X \cup \{f_2\}, w \models af'_2(x)\}$ is inconsistent, then $\{X \setminus \{f_1\} \mid X \in Cand Pgm_{af}(w, \mathcal{I}_1)\} = \{X \setminus \{f_2\} \mid X \in Cand Pgm_{af}(w, \mathcal{I}_2)\}$.*
2. *If $\{X \setminus \{f_1\} \mid X \in Cand Pgm_{af}(w, \mathcal{I}_1)\} = \{X \setminus \{f_2\} \mid X \in Cand Pgm_{af}(w, \mathcal{I}_2)\}$ then $\{x \mid x \in X \cup \{f_1\}, w \models af'_1(x)\}$ is inconsistent $\Leftrightarrow \{x \mid x \in X \cup \{f_2\}, w \models af'_2(x)\}$ is inconsistent.*

Proof (If) Suppose BWOC that for all $w \in \mathcal{W}^I_{EM}(\mathcal{I}_1)$ and all $X \subseteq \Pi_{AM}(w)$; if $\{x \mid x \in X \cup \{f_1\}, w \models af'_1(x)\}$ is inconsistent iff $\{x \mid x \in X \cup \{f_2\}, w \models af'_2(x)\}$ is inconsistent, but there exists w s.t.
$\{X \setminus \{f_1\} \mid X \in Cand Pgm_{af}(w, \mathcal{I}_1)\} \neq \{X \setminus \{f_2\} \mid X \in Cand Pgm_{af}(w, \mathcal{I}_2)\}$. However, the pre-condition of this statement implies that $\{X \setminus \{f_1\} \mid X \subseteq Cand Pgm_{af}(w, \mathcal{I}_1)\} = \{X \setminus \{f_2\} \mid X \subseteq Cand Pgm_{af}(w, \mathcal{I}_2)$ which gives us a contradiction.
(Only-If) Suppose BWOC that for all w, $\{X \setminus \{f_1\} \mid X \in Cand Pgm_{af}(w, \mathcal{I}_1)\} = \{X \setminus \{f_2\} \mid X \in Cand Pgm_{af}(w, \mathcal{I}_2)\}$, but there exists a world $w \in \mathcal{W}^I_{EM}(\mathcal{I}_1)$ and set $X \subseteq \Pi_{AM}(w)$ s.t. exactly one of $\{x \mid x \in X \cup \{f_1\}, w \models af'_1(x)\}$, $\{x \mid x \in X \cup \{f_2\}, w \models af'_2(x)\}$ is inconsistent. As a first case, let us assume that $\Pi_{AM}(w) \cup \{f_1\}$ is consistent. As $\{X \setminus \{f_1\} \mid X \in Cand Pgm_{af}(w, \mathcal{I}_1)\} = \{X \setminus \{f_2\} \mid X \in Cand Pgm_{af}(w, \mathcal{I}_2)\}$, this implies that $\Pi_{AM}(w) \cup \{f_2\}$ must also be consistent as each of those sets must then have exactly one element. In this case a contradiction arises, hence both $\Pi_{AM}(w) \cup \{f_1\}$, $\Pi_{AM}(w) \cup \{f_2\}$ must be classically inconsistent. Now let us consider the other case. As $\Pi_{AM}(w)$ is consistent and

all its subsets are consistent, then we must consider some $X \subseteq \Pi_{AM}(w)$ where $X \cup \{f_1\}$ is not consistent. Hence, $X \cup \{f_2\}$ must be consistent. As $\Pi_{AM}(w) \in CandPgm_{af}(w, \mathcal{I}_2)$, we know that $X \in \{X \setminus \{f_2\} \mid X \in CandPgm_{af}(w, \mathcal{I}_2)\}$ iff $X \cup \{f_2\}$ is consistent, so it must be in that set. However, $X \notin \{X \setminus \{f_1\} \mid X \in CandPgm_{af}(w, \mathcal{I}_1)\}$ as $X \cup \{f_1\}$ is not consistent — this is a contradiction. □

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. J. Sym. Log. **50**(2), 510–530 (1985)
2. Altheide, C.: Digital Forensics with Open Source Tools. Syngress (2011)
3. Bondarenko, A., Dung, P.M., Kowalski, R.A., Toni, F.: An abstract, argumentation-theoretic approach to default reasoning. Artif. Intell. **93**(1), 63–101 (1997)
4. Chesñevar, C.I., Simari, G.R., Alsinet, T., Godo, L.: A logic programming framework for possibilistic argumentation with vague knowledge. In: Proceedings of UAI 2004, pp. 76–84 (2004)
5. Corp., S.: Stuxnet 0.5: Disrupting Uranium Processing at Natanz. Symantec Connect (2013). http://www.symantec.com/connect/blogs/stuxnet-05-disrupting-uranium-processing-natanz
6. Doyle, J.: A truth maintenance system. Artif. Intell. **12**(3), 231–272 (1979)
7. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. **77**, 321–357 (1995)
8. Dunne, P.E., Hunter, A., McBurney, P., Parsons, S., Wooldridge, M.: Weighted argument systems: basic definitions, algorithms, and complexity results. Artif. Intell. **175**(2), 457–486 (2011)
9. Falappa, M.A., García, A.J., Kern-Isberner, G., Simari, G.R.: On the evolving relation between belief revision and argumentation. Knowl. Eng. Rev. **26**(1), 35–43 (2011). doi:10.1017/S0269888910000391
10. Falappa, M.A., Kern-Isberner, G., Reis, M., Simari, G.R.: Prioritized and non-prioritized multiple change on belief bases. J. Philosophical Logic **41**(1), 77–113 (2012)
11. Falappa, M.A., Kern-Isberner, G., Simari, G.R.: Explanations, belief revision and defeasible reasoning. Artif. Intell. **141**(1/2), 1–28 (2002)
12. Falappa, M.A., Kern-Isberner, G., Simari, G.R.: Argumentation in artificial intelligence, chap. In: Rahwan, I., Simari, G.R. (eds.) Belief Revision and Argumentation Theory, pp. 341–360. Springer (2009)
13. Falliere, N., Murchu, L.O., Chien, E.: W32.Stuxnet Dossier Version 1.4. Symantec Corporation (2011)
14. Fazzinga, B., Flesca, S., Parisi, F.: On the complexity of probabilistic abstract argumentation. In: Proceedings of IJCAI 2013, pp. 898–904 (2013)
15. García, A.J., Simari, G.R.: Defeasible logic programming: an argumentative approach. TPLP **4**(1–2), 95–138 (2004)
16. Gardenfors, P.: Knowledge in flux: modeling the dynamics of epistemic states. MIT Press, Cambridge (1988)
17. Gärdenfors, P.: Belief revision, vol. 29. Cambridge University Press (2003)
18. Gottlob, G., Lukasiewicz, T., Martinez, M.V., Simari, G.I.: Query answering under probabilistic uncertainty in Datalog+/– ontologies. AMAI, 37–72 (2013)
19. Haenni, R., Kohlas, J., Lehmann, N.: Probabilistic argumentation systems. Springer (1999)
20. Hansson, S.: Semi-revision. J. App. Non-Classical Logics **7**(1–2), 151–175 (1997)
21. Hansson, S.O.: Kernel contraction. J. Symb. Log. **59**(3), 845–859 (1994)
22. Heuer, R.J.: Psychology of Intelligence Analysis. Center for the Study of Intelligence (1999). http://www.odci.gov/csi/books/19104/index.html
23. Hunter, A.: Some foundations for probabilistic abstract argumentation. In: Proceedings of COMMA 2012, pp. 117–128 (2012)
24. Hunter, A.: A probabilistic approach to modelling uncertain logical arguments. Int. J. Approx. Reasoning **54**(1), 47–81 (2013)
25. Khuller, S., Martinez, M.V., Nau, D.S., Sliva, A., Simari, G.I., Subrahmanian, V.S.: Computing most probable worlds of action probabilistic logic programs: scalable estimation for $10^{30,000}$ worlds. AMAI **51**(2–4), 295–331 (2007)
26. Krause, P., Ambler, S., Elvang-Gørannson, M., Fox, J.: A logic of argumentation for reasoning under uncertainty. Comput. Intell. **11**(1), 113–131 (1995)
27. Langner, R.: Matching Langner Stuxnet analysis and Symantic dossier update. Langner Communications GmbH (2011). http://www.langner.com/

28. Li, H., Oren, N., Norman, T.J.: Probabilistic argumentation frameworks. In: Proceedings of TAFA, pp. 1–16 (2011)
29. Lloyd, J.W.: Foundations of Logic Programming, 2nd edn. Springer (1987)
30. Martinez, M.V., García, A.J., Simari, G.R.: On the use of presumptions in structured defeasible reasoning. In: Proceedings of COMMA, pp. 185–196 (2012)
31. Modgil, S., Prakken, H.: A general account of argumentation with preferences. Artif. Intell. **195**, 361–397 (2013)
32. Nilsson, N.J.: Probabilistic logic. Artif. Intell. **28**(1), 71–87 (1986)
33. Prakken, H.: An abstract framework for argumentation with structured arguments. Argument and Computation **1**, 93–124 (2010)
34. Rahwan, I., Simari, G.R.: Argumentation in Artificial Intelligence. Springer (2009)
35. Richardson, M., Domingos, P.: Markov logic networks. Mach. Learn. **62**(1–2), 107–136 (2006)
36. Riley, L., Atkinson, K., Payne, T., Black, E.: An implemented dialogue system for inquiry and persuasion. In: Theory and Applications of Formal Argumentation, Lecture Notes in Computer Science, pp. 67–84. Springer, Berlin (2011)
37. Shadows in the Cloud: Investigating Cyber Espionage 2.0. Tech. rep., Information Warfare Monitor & Shadowserver Foundation (2010)
38. Shafer, G., et al.: A mathematical theory of evidence, vol. 1. Princeton University Press, Princeton (1976)
39. Shakarian, P., Shakarian, J., Ruef, A.: Introduction to Cyber-Warfare: A Multidisciplinary Approach. Syngress (2013)
40. Shakarian, P., Simari, G.I., Falappa, M.A.: Belief revision in structured probabilistic argumentation. In: Proceedings of Foundations of Information and Knowledge Systems, pp. 324–343 (2014)
41. Shakarian, P., Simari, G.I., Moores, G., Parsons, S., Falappa, M.A.: An argumentation-based framework to address the attribution problem in cyber-warfare. In: Proceedings of Cyber Security (2014)
42. Simari, G.I., Martinez, M.V., Sliva, A., Subrahmanian, V.S.: Focused most probable world computations in probabilistic logic programs. AMAI **64**(2–3), 113–143 (2012)
43. Simari, G.R., Loui, R.P.: A mathematical treatment of defeasible reasoning and its implementation. Artif. Intell. **53**(2–3), 125–157 (1992)
44. Spitzner, L.: Honeypots: catching the insider threat. In: Proceedings of ACSAC 2003, pp. 170–179. IEEE Computer Society (2003)
45. Stolzenburg, F., García, A., Chesñevar, C.I., Simari, G.R.: Computing generalized specificity. J Non-Classical Logics **13**(1), 87–113 (2003)
46. Thimm, M.: A probabilistic semantics for abstract argumentation. In: Proceedings of ECAI 2012, pp. 750–755 (2012)
47. Thonnard, O., Mees, W., Dacier, M.: On a multicriteria clustering approach for attack attribution. SIGKDD Explor. **12**(1), 11–20 (2010)