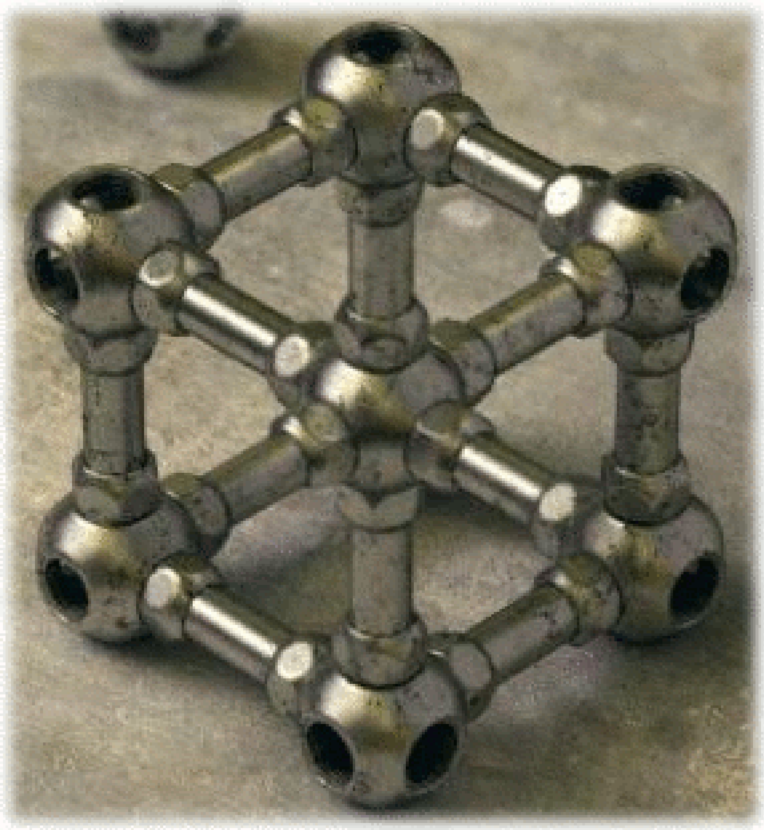

Satisfiability in $\mathcal{EL}$ with sets of Probabilistic ABoxes [1]

Author(s):

Marcelo Finger
Renata Wassermann
Fabio G. Cozman

[1] This work was supported by Fapesp Project LogProb, grant 2008/03995-5, São Paulo, Brazil.

Satisfiability in $\mathcal{EL}$ with sets of Probabilistic ABoxes*

Marcelo Finger, Renata Wassermann, and Fabio G. Cozman

University of São Paulo, São Paulo, Brazil
mfinger@ime.usp.br,renata@ime.usp.br,fgcozman@usp.br

Abstract. We present $\mathcal{EL}$PA, a probabilistic extension of the lightweight DL $\mathcal{EL}$ with a fixed TBox and a set of probabilistic ABoxes, and study the problem of satisfiability in such context.

1 Introduction

This work studies an extension of Description Logic $\mathcal{EL}$, called $\mathcal{EL}$PA, that allows for probabilistic assessments on ABoxes. We concentrate on the problem of verifying the *satisfiability* of an $\mathcal{EL}$PA-knowledge base, proposing algorithms for this problem based on recent advances on probabilistic satisfiability (PSAT) [FB11]. Consider an example, adapted from [LS10].

Example 1.1 Two symptoms of Lyme disease are fever and fatigue. As these symptoms are common and the disease is rare, the chance that they are indeed caused by Lyme disease is small. Nevertheless, because the disease is of difficult diagnosis, patients get treated if there is a chance that they have it.

The TBox $\mathcal{T}_0$ contains the following axioms:	And the following ABox $\mathcal{A}_0$:
Fatigue $\sqsubseteq$ Symptom	Fever(s_1)
Fever $\sqsubseteq$ Symptom	hasSymptom(john, s_1)
Lyme $\sqsubseteq$ Disease	Fatigue(s_2)
Symptom $\sqsubseteq$ $\exists$hasCause.Disease	hasSymptom(john, s_2)
Patient $\sqsubseteq$ $\exists$suspectOf.Disease	Patient(john)
Patient $\sqsubseteq$ $\exists$hasSymptom.Symptom	
$\exists$hasSymptom.($\exists$hasCause.Lyme)$\sqsubseteq$ $\exists$suspectOf.Lyme	

Consider also the following probabilistic statements originating from medical experience on symptoms that are caused by Lyme disease.

$\mathcal{A}_1 = \exists\text{hasCause.Lyme}(s_1),\ P(\mathcal{A}_1) \geq 0.1$

$\mathcal{A}_2 = \exists\text{hasCause.Lyme}(s_2),\ P(\mathcal{A}_2) \geq 0.2$

Now we want to know whether we can consistently assert an upper bound p_{ub} for the probability of John having Lyme disease:

$\mathcal{A}_3 = \exists\text{suspectOf.Lyme(john)},\ P(\mathcal{A}_3) \leq p_{\text{ub}}$ □

The set of statements in the example above is a *probabilistic knowledge base.* It contains four probability assignments; $\mathcal{A}_0$, which is the conjunction of 5 atomic statements, is assigned probability 1; the other three atomic ABoxes $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$ get probabilities smaller than 1.

* Work supported by Fapesp Thematic Project 2008/03995-5 (LOGPROB). Authors supported by CNPq grants PQ 302553/2010-0, 304043/2010-9, 305395/2010-6.

Our method is an alternative to existing combinations of DL with probabilities that impose deterministic restrictions on probabilities. For example, [LS10] assigns probabilities to concepts over a fixed interpretation, forcing the probability of ABoxes to be either 0 or 1. No such deterministic "side effects" occur in our method.

Our goal in this work is to formally define the notion of $\mathcal{EL}$PA-knowledge bases and its satisfiability problem and provide algorithms to verify it. Adding probabilities to logic sentences usually adds complexity; e.g. probabilistic 2SAT is NP-complete [GKP88]. We show that $\mathcal{EL}$PA-satisfiability is in NP.

1.1 Related work

There have been several proposals to add probabilities to description logics in recent years [LS08]; most of this work is based on various kinds of probabilistic logic [GT07,Hal03].

Probabilistic description logics differ in several dimensions. Some approaches associate probabilities with elements of TBoxes [Jae94,KLP97,CP09], while others associate probabilities with ABoxes [DS05], and still others combine both kinds of assessments [Luk08]. In this paper we focus on probabilities over ABoxes.

As an alternative classification scheme, some logics assign probabilities over elements of the domain [DPP06,DS05,Hei94,Jae94,KLP97,Luk08], while others assign probabilities over interpretations [CL06,DFL08,LS10], with some logics in between [Seb94]. In this paper we focus on probabilities over interpretations.

Yet another classification is possible, as we have probabilistic description logics that allow for assessments of stochastic independence, often organized through graphs [CL06,DPP06,KLP97,CP09], and logics that do not allow for assessments of stochastic independence [DS05,Hei94,Jae94,Luk08,LS10]. In this paper we do not allow for stochastic independence.

In a sense, our work is a refinement of First Order Probabilistic Logic by Jaumard et al [JFSS06]; however, we use the decidable and tractable logic $\mathcal{EL}$, and we show that our probabilistic version remains in NP. Note that probability assignments remain *external* to the logic $\mathcal{EL}$; this has the advantage of making it capable of dealing with conditional probabilities of ABox statements in a classical manner, as $P(A(a)|B(b)) = P(A(a) \wedge B(b))/P(B(b)$. A complete treatment of conditional probabilities remains outside the scope of this work. Another related problem is *probability inference*; that is, determining the maximal and minimal values for p_{ub} that leave the knowledge base satisfiable; this problem is also outside the scope of this work.

1.2 Organization of the paper

The remainder of the paper is organized as follows. In the next section, we introduce the logic $\mathcal{EL}$PA. We first define the probabilistic assignments that are allowed in the ABox, and then formalise the satisfiability problem for $\mathcal{EL}$PA, showing that it is in NP. We show that the probabilistic knowledge base in

$\mathcal{EL}$PA can be translated into a normal form that is used in Section 3, where an algorithm for testing the satisfiability of $\mathcal{EL}$PA is presented.

2 The Probabilistic DL $\mathcal{EL}$PA

We introduce $\mathcal{EL}$PA, a probabilistic extension of the polynomial-time Description Logic $\mathcal{EL}$ with a fixed TBox and a set of probabilistic ABoxes.

We first establish a regular $\mathcal{EL}$ vocabulary. Fix countably infinite sets $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$ of *concept names*, *role names*, and *individual names*, respectively. The set of $\mathcal{EL}$-*concepts* is given by the following syntax rules:

$$C ::= A \mid C \sqcap D \mid \exists r.C$$

where A ranges over $\mathsf{N_C}$, C and D over $\mathcal{EL}$-concepts and r over $\mathsf{N_R}$. No negation or disjunction of concepts is expressible in this language.

A *TBox* is a finite set of *concept inclusions* (CIs) of the form $C \sqsubseteq D$; TBoxes usually represent an ontology. On the other hand, *ABoxes* represent instance data and obey the following syntax rules

$$\mathcal{A} ::= C(a) \mid r(a,b) \mid \mathcal{A} \wedge \mathcal{A}'$$

where C and r are as before, $a, b \in \mathsf{N_I}$ and $\mathcal{A}$, $\mathcal{A}'$ range over ABoxes.

The standard $\mathcal{EL}$ semantics is used for TBoxes and ABoxes, based on interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set called *domain* and $\cdot^{\mathcal{I}}$ is an *interpretation function* that maps each each $A \in \mathsf{N_C}$ to a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, each $r \in \mathsf{N_R}$ to a subset $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and each $a \in \mathsf{N_I}$ to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$; see [BBL05]. We extend the interpretation $\mathcal{I}$ for all concepts in the usual way. So $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$ and $(\exists r.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | \exists y \in \Delta^{\mathcal{I}} : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$. Then concept inclusion $C \sqsubseteq D$ is satisfied by interpretation $\mathcal{I}$, represented by $\mathcal{I} \models C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Similarly TBox $\mathcal{T}$ is satisfied by interpretation $\mathcal{I}$, $\mathcal{I} \models \mathcal{T}$, iff $\mathcal{I} \models C \sqsubseteq D$ for every $C \sqsubseteq D \in \mathcal{T}$.

For ABoxes, we say that $C(a)$ is satisfied by $\mathcal{I}$, represented by $\mathcal{I} \models C(a)$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$; similarly, $\mathcal{I} \models r(a,b)$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$; and $\mathcal{I} \models \mathcal{A} \wedge \mathcal{A}'$ if both $\mathcal{I} \models \mathcal{A}$ and $\mathcal{I} \models \mathcal{A}'$. If both a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ are true under $\mathcal{I}$, we say that the pair $(\mathcal{T}, \mathcal{A})$, called a *(deterministic) knowledge base*, is *satisfied* by $\mathcal{I}$. The problem of deciding if a deterministic knowledge base $(\mathcal{T}, \mathcal{A})$ is satisfiable in $\mathcal{EL}$ can be solved in polynomial time [Bra04].

2.1 Probability Assignments to ABoxes

We now introduce probabilities in ABoxes. We deal with probability distribution π over the set of interpretations endowed with a suitable algebra.

Let $\mathcal{T}$ be a TBox. Given a probability distribution π over the set of all interpretations $\mathbb{I}$, the probability of an ABox $\mathcal{A}$ in the context of $\mathcal{T}$ is given by the probability of all interpretations that satisfy all of $\mathcal{T}$ and $\mathcal{A}$, that is, the probability of $\{\mathcal{I} | \mathcal{I} \models \mathcal{A} \text{ and } \mathcal{I} \models \mathcal{T}\}$.

A *probability assignment* to an ABox, is an expression of the form

$$P(\mathcal{A}) \bowtie p,$$

where $\mathcal{A}$ is an ABox, $\bowtie \in \{\leq, \geq, =\}$ and $p \in \mathbb{Q}, 0 \leq p \leq 1$. Note that probability assignments are external to the logic $\mathcal{EL}$, and are *not* statements in the logic.

Let $\mathcal{PA}$ be a set of k probability assignments

$$\mathcal{PA} = \{P(\mathcal{A}_i) \bowtie_i p_i | 1 \leq i \leq k\}.$$

Then the pair $\langle \mathcal{T}, \mathcal{PA} \rangle$ is a *probabilistic knowledge base* in the DL $\mathcal{EL}$ with sets of probability assignments, $\mathcal{EL}$PA. Clearly, all probability assignments in $\mathcal{PA}$ are to be evaluated in the context $\mathcal{T}$.

The main problem of this work is, given an $\mathcal{EL}$PA probabilistic knowledge base, determine whether there exists a probability distribution π such that, in the context of the TBox $\mathcal{T}$, π satisfies all the assignments in $\mathcal{PA}$; this is the *satisfiability problem* for an $\mathcal{EL}$PA-knowledge base. Before we formalize this problem, we must "finitize" the set of all interpretations of a TBox $\mathcal{T}$, as the set $\mathbb{I}_\mathcal{T}$ of all interpretations of a TBox $\mathcal{T}$ is uncountably infinite.

For that, define an equivalence relation $\simeq_{\mathcal{PA}} \subseteq \mathbb{I}_\mathcal{T} \times \mathbb{I}_\mathcal{T}$, where $\mathcal{PA} = \{P(\mathcal{A}_i) = p_i | 1 \leq i \leq k\}$ is a set of probabilistic assignments, such that

$$\mathcal{I} \simeq_{\mathcal{PA}} \mathcal{I}' \text{ iff for every } k, 1 \leq i \leq k, \mathcal{I} \models \mathcal{A}_i \text{ if and only if } \mathcal{I}' \models \mathcal{A}_i;$$

that is, $\mathcal{I}$ and $\mathcal{I}'$ satisfy the same ABoxes in $\mathcal{PA}$ in a context of TBox $\mathcal{T}$.

Lemma 2.1 *Let n be the number of atomic elements in $\mathcal{PA}$. The relation $\simeq_{\mathcal{PA}}$ is an equivalence relation on $\mathbb{I}_\mathcal{T} \times \mathbb{I}_\mathcal{T}$ and the set of equivalence classes $\mathbb{I}_\mathcal{T}/{\simeq_{\mathcal{PA}}}$ has at most 2^n distinct equivalence classes.*

Each equivalence class of $\mathbb{I}/{\simeq_{\mathcal{PA}}}$ will be represented by any interpretation $\mathcal{I}$ in it. We can now formalise the satisfiability problem for an $\mathcal{EL}$PA-knowledge base $\langle \mathcal{T}, \mathcal{PA} \rangle$. We will write $\mathcal{I}(\mathcal{A}) = 1$ for $\mathcal{I} \models \mathcal{A}$ and $\mathcal{I}(\mathcal{A}) = 0$ for $\mathcal{I} \not\models \mathcal{A}$.

2.2 The Satisfiability of Probabilistic Knowledge Bases

Let n be the number of atomic elements in $\mathcal{PA}, |\mathcal{PA}| = k$. Consider $\mathcal{I}_1, \ldots, \mathcal{I}_{n'}$, $n' \leq 2$ be all the $\simeq_{\mathcal{PA}}$-distinct interpretations that satisfy $\mathcal{T}$. Consider a $k \times n'$ matrix $A = [a_{ij}]$ such that $a_{ij} = \mathcal{I}_j(\mathcal{A}_i)$. The *probabilistic satisfiability problem* for an $\mathcal{EL}$PA-knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{PA} \rangle$ is to decide if there is a probability vector π of dimension n' that obeys the *$\mathcal{EL}$PA-restrictions*:

$$A\pi \bowtie p, \qquad \sum \pi_i = 1, \qquad \pi \geq 0. \tag{1}$$

An $\mathcal{EL}$PA-knowledge base $\mathcal{K}$ is *satisfiable* iff its associated $\mathcal{EL}$PA-restrictions (1) have a solution. If π is a solution to (1) we say that π satisfies $\mathcal{K}$. The last two conditions of (1) force π to be a probability distribution. It is convenient to assume that first two conditions of (1) are joined, A is a $(k+1) \times n'$ matrix with 1's at its first line, $p_1 = 1$ in vector $p_{(k+1)\times 1}$, so $\bowtie_1$-relation is "="; we will keep this convention in the rest of the paper.

Example 2.2 Recall Example 1.1 with $p_{\text{ub}} = 0.3$, where the probability of John having Lyme is at most 30%. We consider only the 3 ABox with probabilistic assignments, and only interpretations of these atoms that are jointly consistent with the fixed TBox and the fixed (probability 1) ABox formulas. Consider the following probability distribution π and the probability it assigns to the ABoxes in Example 1.1.

	π	$\mathcal{A}_1$	$\mathcal{A}_2$	$\mathcal{A}_3$
$\mathcal{I}_1$	0.75	0	0	0
$\mathcal{I}_2$	0.10	0	1	1
$\mathcal{I}_3$	0.03	1	0	1
$\mathcal{I}_4$	0.12	1	1	1
	1.00	0.15	0.22	0.25

It is easy to verify that the interpretations $\mathcal{I}_1$–$\mathcal{I}_4$ are all consistent with $\langle \mathcal{T}_0, \mathcal{A}_0 \rangle$, so all probability relations are verified by π, so probabilistic database for $p_{\text{ub}} = 0.3$ is satisfiable. □

Some important questions remain: how to compute a probability distribution when one exists, and whether that probabilistic knowledge base remains satisfiable when $p_{\text{ub}} = 0.05$ or not, and how to verify it. This paper presents algorithms for that.

An important result of [GKP88] guarantees that a satisfiable knowledge base has a "small" witness:

Fact 2.3 *If $\mathcal{ELPA}$-restrictions (1) for knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{PA} \rangle$ with $\mathcal{PA} = \{P(\mathcal{A}_i) = p_i | 1 \leq i \leq k\}$ have a solution, then there are $k+1$ columns of matrix A such that the system $A_{(k+1)\times(k+1)}\pi = p_{(k+1)\times 1}$ has a solution $\pi \geq 0$.*

This result is a consequence of Caratheodory's Theorem [Eck93], which states that if a k-dimensional point is a convex combination of m points, then it is a convex combination of at most $k+1$ points among them. Fact 2.3 gives an NP-certificate for the satisfiability of an $\mathcal{EL}$PA-knowledge basel; hence:

Corollary 2.4 *The $\mathcal{EL}$PA-satisfiability problem is in NP.*

Finding polynomial-sized certificates is the heart of the matter. We will now study algorithms that solve the $\mathcal{EL}$PA-satisfiability problem. We start by defining a normal form for $\mathcal{EL}$PA-knowledge base. Note that Fact 2.3 is stated for equality only, and we also allow inequalities; the normal form will be useful both for the algorithms and for showing that all those cases can be reduced to equality only, with all probabilistic assignments over atoms only.

2.3 A Normal Form for Probabilistic Knowledge Base

For the sake of providing a normal form, we add a few new convenient definitions. Let N_0 be a set of 0-ary atomic propositions. A *propositional rule* is an expression of the form $q \to \mathcal{A}_1$ or $\mathcal{A}_2 \to q$, where $q \in \mathsf{N_0}$ and $\mathcal{A}_1, \mathcal{A}_2$ ABoxes, with the obvious semantic that $\mathcal{I} \models q \to \mathcal{A}_1$ iff $\mathcal{I} \not\models q$ or $\mathcal{I} \models \mathcal{A}_1$; and $\mathcal{I} \models \mathcal{A}_2 \to q$ iff $\mathcal{I} \models q$ or $\mathcal{I} \not\models \mathcal{A}_2$. We extend the notion of ABox such that

$$\mathcal{A} ::= C(a) \mid r(a,b) \mid q \mid \mathcal{A} \wedge \mathcal{A}'$$

such that $q \in \mathsf{N_0}$ and $C(a), r(a,b), \mathcal{A}, \mathcal{A}'$ are as before; we call $q, C(a), r(a,b)$ *atomic* ABoxes.

If $\mathcal{R}$ is a set of propositional rules and $\mathcal{A}$ an ABox, $\mathcal{R} \cup \mathcal{A}$ is a set of Horn clauses, and thus has a polynomial-time computable minimal model; so the $\mathcal{I}$-satisfiability of $\mathcal{R} \cup \mathcal{A}$ reduces to the $\mathcal{I}$-satisfiability of the atomic positive formulas in its minimal model. Thus the satisfiability problem of $\mathcal{EL}$ with TBoxes, ABoxes and sets of propositional rules can be achieved in polynomial time.

We also distinguish deterministic ABoxes, which are assigned probability 1, from probabilistic ABoxes, which are assigned probabilities < 1.

We then extend previous definitions with the notion of a set of propositional rules. For the rest of this paper, an *extended* $\mathcal{EL}$PA-knowledge base is a 4-tuple $\mathcal{K}^e = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{PA} \rangle$, in which the *probabilistic assignment of ABoxes* $\mathcal{PA}$ is evaluated in an *(deterministic) evaluation context* consisting of a triple $\mathcal{C} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ of TBox, propositional rules and deterministic ABox $\mathcal{A}$; we also represent $\mathcal{K}^e = \langle \mathcal{C}, \mathcal{PA} \rangle$. Clearly, an $\mathcal{EL}$PA-knowledge base $\mathcal{K}$ is a special case of an extended $\mathcal{EL}$PA-knowledge base $\mathcal{K}^e$ the previous view in which $\mathcal{R} = \varnothing$ and $\mathcal{A}$ is part of $\mathcal{PA}$.

Now we define the normal form. A knowledge base $\mathcal{K}^e = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{PA} \rangle$ is in *(atomic) normal form* if $\mathcal{PA}$ is of the form

$$\mathcal{PA} = \{P(y_i) = p_i \mid y_i \text{ is an atom}, 1 \le i \le k\}, \text{ with } 0 < p_i < 1.$$

In this case, $\mathcal{PA}$ is an *atomic probability assignment* evaluated in context $\mathcal{C} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$. Clearly, $\mathcal{C}$ is a small generalisation of the deterministic knowledge bases of, for instance, [BBL05].

By adding a small number of propositional rules, any knowledge base can be brought into atomic normal form.

Theorem 2.5 (Normal Form) *For every extended $\mathcal{EL}$PA-knowledge base $\mathcal{K}^e$ there exists an atomic normal form knowledge base $\mathcal{K}^e_{nf}$ that is satisfiable iff $\mathcal{K}^e$ is; the former can be obtained from the latter in polynomial time $O(k \times \ell)$, where $k = |\mathcal{PA}|$ and ℓ is the largest number of conjuncts in an ABox in $\mathcal{PA}$.* □

Example 2.6 We transform the knowledge base(s) of Example 1.1 into the normal form. The TBox $\mathcal{T}_0$ and deterministic ABox $\mathcal{A}_0$ remain the same. We introduce atoms q_1, q_2, q_3 for ABoxes $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ respectively, which generates the following set or rules $\mathcal{R}_0$:

$$q_1 \to \exists\text{hasCause.Lyme}(s_1), \quad q_2 \to \exists\text{hasCause.Lyme}(s_2), \quad \exists\text{suspectOf.Lyme(john)} \to q_3$$

$\mathcal{C}_0 = \langle \mathcal{T}_0, \mathcal{R}_0, \mathcal{A}_0 \rangle$ is the evaluation context; the atomic probability assignment is

$$\mathcal{PA}_0 = \{ \quad P(q_1) = 0.1, \qquad P(q_2) = 0.2. \qquad P(q_3) = p_{\text{ub}} \quad \}.$$

The normal form knowledge base is $\mathcal{K}_0^e = \langle \mathcal{C}_0, \mathcal{PA}_0 \rangle$. Note that the probability distribution of Example 2.2 does not satisfy $\mathcal{K}_0^e$ when $p_{\text{ub}} = 0.3$, but by Theorem 2.5 there must exist other interpretations involving q_1, q_2, q_3 and rules $\mathcal{R}_0$ and another probability distribution π that satisfies $\mathcal{K}_0^e$. $\square$

The following result allows us to see a satisfiable normal form knowledge base $\mathcal{K}_{nf}^e$ as an interaction between a solution to assignments $\mathcal{PA}$ constrained by the $\mathcal{EL}$-decisions of context $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$. An interpretation $\mathcal{I}$ is $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$-consistent if $\mathcal{I}$ jointly satisfies $\mathcal{T}$, $\mathcal{R}$ and $\mathcal{A}$. Recall that we represent the binary $\mathcal{I}$-evaluation of ABoxes such that $\mathcal{I}(\mathcal{A}) = 1$ iff $\mathcal{I} \models \mathcal{A}$. Lemma 2.7 is the basis for the $\mathcal{EL}$PA-satisfiability solving algorithm that we present in the next section.

Lemma 2.7 *A normal form knowledge base $\mathcal{K}^e = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{PA} \rangle$ is satisfiable iff there is a binary $(k+1) \times (k+1)$-matrix $A_{\mathcal{K}^e}$, such that all of its $\{0,1\}$-columns represent interpretations that are $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$-consistent and $A_{\mathcal{K}^e} \cdot \pi = p$ has a solution $\pi \geq 0$.*

3 An Algorithm for $\mathcal{EL}$PA Satisfiability

We present a logic-algebraic algorithm to verify the satisfiability of a normal form knowledge base $\mathcal{K}^e = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{PA} \rangle$ and, if the answer is positive, present a satisfying model in the form of a set of $k+1$ $\mathcal{EL}$-interpretations, where $k = |\mathcal{PA}|$, and a probability distribution over them.

We first establish some terminology. If A is a matrix, A^j is its j-th column and A_i is its i-th line; $A_{(s)}$ is the state of matrix A at step s. If A is a matrix and b a column of compatible dimension, $A[j := b]$ is obtained by replacing A's j-th column with b. A square matrix that has an inverse is *non-singular*. A matrix A that satisfies conditions (2) is a *feasible solution* for $\mathcal{PA}$.

$$\begin{bmatrix} 1 & \cdots & 1 \\ a_{1,1} & \cdots & a_{1,k+1} \\ \vdots & \ddots & \vdots \\ a_{k,1} & \cdots & a_{k,k+1} \end{bmatrix} \cdot \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_{k+1} \end{bmatrix} = \begin{bmatrix} 1 \\ p_1 \\ \vdots \\ p_k \end{bmatrix}, \qquad a_{i,j} \in \{0,1\}, \quad A \text{ is non-singular}, \quad \pi_j \geq 0 \tag{2}$$

We always assume that the lines of A are ordered such that the input probabilities $p_1, \ldots, p_k$ in (2) are in decreasing order. By Lemma 2.7, $\langle \mathcal{C}, \mathcal{PA} \rangle$ has a solution iff there is a partial solution A satisfying (2) such that if $\pi_j > 0$ then $a_{1,j}, \ldots, a_{k,j}$ represent $\mathcal{C}$-consistent interpretations for $1 \leq i \leq k, 1 \leq j \leq k+1$. We usually abuse terminology calling A^j a $\mathcal{C}$-consistent column.

This method is based on PSAT-solving method of [FB11], which is an improvement on the methods of [KP90,HJ00]; it consists of an algebraic optimisation problem in the form of a special linear program of the form

$$\begin{array}{ll} \text{minimize} & \textit{objective function}\langle |J|, f \rangle \\ \text{subject to} & A\pi = p, \pi \geq 0, f = \sum_{j \in J} \pi_j \text{ and} \\ & J = \{j | A^j \text{ is } \mathcal{C}\text{-inconsistent}, \pi_j > 0\} \end{array} \tag{3}$$

which is solved iteratively by the simplex algorithm [BT97]. Matrix A is a $(k+1) \times (k+1)$ {0,1}-matrix, whose columns represents an $\mathcal{EL}$-interpretations and whose lines represent the atoms occurring in $\mathcal{PA}$. An iterative step s receives a matrix $A_{(s)}$ and employs a *column generation* method that solves an *auxiliary problem*; the latter is a logic-based satisfiability problem that employs $\mathcal{EL}$-decision procedure, generates a column that replaces some column in $A_{(s)}$, obtains $A_{(s+1)}$ and decreases the objective function $\langle |J|, f\rangle$, where $\langle |J_1|, f_1\rangle > \langle |J_2|, f_2\rangle$ iff $0 \leq |J_1| < |J_2|$ or $|J_1| = |J_2|$ and $f_1 < f_2$, until its minimum is reached. The objective function is discussed in Section 3.1.

In the iterative method, some columns are not $\mathcal{C}$-consistent and the process is done such that the number of $\mathcal{C}$-consistent columns A^j associated to $\pi_j > 0$ never decreases.

We now define $A_{(0)}$, the starting feasible solution. For that, consider an empty context $\mathcal{C} = \varnothing$, that is a knowledge base $\langle \varnothing, \mathcal{PA}\rangle$. As the elements of p are in decreasing order, consider the $\{0,1\}$-matrix $I^* = [a_{i,j}]_{1 \leq i,j \leq k+1}$ where $a_{i,j} = 1$ iff $i \leq j$, that is, I^* is all 1's in and above the diagonal, 0's elsewhere. As p is in decreasing order, I^* satisfies $\langle \varnothing, \mathcal{PA}\rangle$ and is called a *relaxed solution* for $\langle \mathcal{C}, \mathcal{PA}\rangle$. Clearly, I^* is a feasible for $\mathcal{PA}$. Make $A_{(0)} = I^*$.

Example 3.1 Consider the form of knowledge base in Example 2.6 with $p_{\text{ub}} = 0.3$ (left) and $p_{\text{ub}} = 0.05$ (right). An initial feasible solution for it is $A_{(0)} \cdot \pi^{(0)} = p$, with atoms ordered in decreasing probability, namely

$$\begin{bmatrix} 1&1&1&1 \\ 0&1&1&1 \\ 0&0&1&1 \\ 0&0&0&1 \end{bmatrix} \cdot \begin{bmatrix} 0.7 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.3 \\ 0.2 \\ 0.1 \end{bmatrix} \begin{matrix} \\ q_3 \\ q_2 \\ q_1 \end{matrix} \quad \Bigg| \quad \begin{bmatrix} 1&1&1&1 \\ 0&1&1&1 \\ 0&0&1&1 \\ 0&0&0&1 \end{bmatrix} \cdot \begin{bmatrix} 0.8 \\ 0.1 \\ 0.05 \\ 0.05 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.2 \\ 0.1 \\ 0.05 \end{bmatrix} \begin{matrix} \\ q_2 \\ q_1 \\ q_3 \end{matrix}$$

On the left, all columns are $\mathcal{C}$-consistent, so the problem is satisfiable with solution $A_{(0)}$ and π as above. On the right, the second and third columns of $A_{(0)}$ are $\mathcal{C}$-inconsistent, so the decision is not yet made.

The relaxed solution is the initial feasible solution of our method. Further feasible solutions are obtained by generating new $\{0,1\}$-columns and substituting them into a feasible solution, as shown by the following.

It is well known from the pivoting in the simplex algorithm that given any $\{0,1\}$-column of the form $b = [1 \ b_1 \ \cdots \ b_k]'$, $A[j := b]$ is a feasible solution. So we simply assume there is a function $merge(A, b)$ that computes it. Our method moves through feasible solutions, at each step generating a column b that decreases the value of the objective function.

3.1 The Objective Function

In a feasible solution A such that $A\pi = p$ and $\pi \geq 0$, some columns may not be $\mathcal{C}$-consistent. Let $J = \{j | A^j$ is $\mathcal{C}$-inconsistent and $\pi_j > 0\}$; J is the set of column indexes in A corresponding to $\mathcal{C}$-inconsistent columns with non-null associated probability; clearly $|J| \leq k+1$. If $J = \varnothing$, we call A a *solution.* As $|J| = 0$ when a solution is found, it is one component of the objective function. However, it is

Algorithm 3.1 ELPA-satisfiability solver

Input: A normal form ELPA knowledge base $\langle \mathcal{T}, \mathcal{R}, \mathcal{A}, \mathcal{PA} \rangle$.
Output: Solution A; or "No", if unsatisfiable.

1: $p := \mathit{sortDecrescent}(\{1\} \cup \{p_i | P(y_i) = p_i \in \mathcal{PA}\}$;
2: $A_{(0)} := I^*$; $s := 0$; compute $\langle |J_{(s)}|, f_{(s)} \rangle$;
3: **while** $\langle |J_{(s)}|, f_{(s)} \rangle \neq \langle 0, 0 \rangle$ **do**
4: $\quad b^{(s)} = \mathit{GenerateColumn}(A_{(s)}, p, \mathcal{C})$;
5: $\quad$ **return** "No" **if** $b_1^{(s)} < 0$; /* instance is unsat */
6: $\quad A_{(s+1)} = \mathit{merge}(A_{(s)}, b^{(s)})$;
7: $\quad$ increment s; compute $\langle |J_{(s)}|, f_{(s)} \rangle$;
8: **end while**
9: **return** $A_{(s)}$; /* PSAT instance is satisfiable */

not guaranteed that, if a solution exists, we can find a sequence of iterations in which $|J|$ decreases at every step s.

The second component of the objective function is the sum of probabilities of $\mathcal{C}$-inconsistent columns, $f = \sum_{j \in J} \pi_j$. Note that f and $|J|$ become 0 at the same time, which occurs iff a positive decision is reached. The simplex algorithm with appropriate column generation ensures that, if there is a solution, it can be obtained with finitely many steps of non-increasing f-values. We thus use a combined objective function $\langle |J|, f \rangle$ ordered lexicographically.

We first try to minimise the number of $\mathcal{C}$-inconsistent columns; if this is not possible, then minimise f, keeping J constant. So a knowledge base instance $\langle \mathcal{C}, \mathcal{PA} \rangle$ associated to program (3) is satisfiable iff the objective function is minimal at $\langle 0, 0 \rangle$.

Assume there is a function $\mathit{GenerateColumn}(A, p, \mathcal{C})$, presented at Section 3.2, that generates a $\mathcal{C}$-consistent column that decreases the objective function, if one exists; otherwise it returns an illegal column of the form $[-1 \ \cdots]$. Algorithm 3.1 presents a method that decides a PSAT instance by solving problem (3).

Algorithm 3.1 starts with a relaxed solution for $\langle \mathcal{C}, \mathcal{PA} \rangle$ (line 2), and via column generation (line 4) generates another feasible solution (line 6), decreasing the objective function, until either the search fails (line 5) or a solution is found; the latter only occurs with the termination of the loop in lines 3–8, when the objective function reaches $\langle 0, 0 \rangle$.

3.2 Column Generation for ELPA

Algorithm 3.1 is, unsurprisingly, almost the same algorithm for PSAT solving in [FB11]; the only difference between the two rests in the column generation method $\mathit{GenerateColumn}(A, p, \mathcal{C})$.

It has been shown in [FB11] that to eliminate a $\mathcal{C}$-inconsistent column A^j associated to $\pi_j > 0$, a new $\mathcal{C}$-consistent column $b = [1 \ y_1 \dots y_k]'$ to substitute A^j must satisfy the set of linear inequalities:

$$(LR_{ij}) \quad (A_j^{-1}\pi_i - A_i^{-1}\pi_j)[1 \ y_1 \dots y_k]' \geq 0, \quad 1 \leq i \leq k+1 \qquad (4)$$

Such a column is here obtained by a combination of a SAT solver, which guarantees that (4) is verified, with an $\mathcal{EL}$-solver to guarantee that b is $\mathcal{C}$-consistent. This combination can be done in several ways.

(a) By coding the polynomial time $\mathcal{EL}$-decision in a SAT solver.
(b) By using $\mathcal{EL}$-theories as an SMT (SAT Modulo Theories) engine.
(c) By coupling an $\mathcal{EL}$-solver at the end of the SAT solver, rejecting $\mathcal{C}$-inconsistent answers, and proceeding with the SAT solver after the rejection.

The latter option is perhaps the most straightforward and is the one we employ here.

Example 3.2 Recall the matrix $A_{(0)}$ in Example 3.1 on the right, whose second and third columns were $\mathcal{C}$ inconsistent. Applying Algorithm 3.1 with column generation as above, all 3 columns generated by the SAT solver were rejected by the $\mathcal{EL}$-solver, so no column could be generated that minimised the objective function in $\langle 0, 0 \rangle$. Therefore the corresponding $\mathcal{EL}$PA-knowledge base is unsatisfiable. □

Theorem 3.3 *Algorithm 3.1 with column generation as above is correct and always terminates.*

4 Conclusions and Further Work

We have introduced the notion of $\mathcal{EL}$PA-knowledge bases and its satisfiability problem, and we have shown that the problem has a finite version that can be tacked by algorithms that resemble PSAT solvers. We have also provided complexity upper bounds for these algorithms.

Algorithm 3.1 has the theoretical possibility of generating an exponential number of steps. It remains an open problem to find an example in which such an exponential number of steps occur. It also remains an open problem whether a polynomial time algorithm exists for ELPA-satisfiability. Our plan for future work is to investigate the practical behavior of our algorithms, and to explore logics that allow for probability over TBoxes and for stochastic independence.

References

[BBL05] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the EL envelope. In *IJCAI05, 19th International Joint Conference on Artificial Intelligence*, pages 364–369, 2005.

[Bra04] Sebastian Brandt. Polynomial time reasoning in a description logic with existential restrictions, gci axioms, and - what else? In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004*, pages 298–302, 2004.

[BT97] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to linear optimization.* Athena Scientific, 1997.

[CP09] FG Cozman and RB Polastro. Complexity analysis and variational inference for interpretation-based probabilistic description logics. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009.

[Eck93] J. Eckhoff. Helly, Radon, and Caratheodory type theorems. In P. M. Gruber and J. M. Wills, editors, *Handbook of Convex Geometry*, pages 389–448. Elsevier Science Publishers, 1993.

[FB11] Marcelo Finger and Glauber De Bona. Probabilistic satisfiability: Logic-based algorithms and phase transition. In *To appear on IJCAI2011*, 2011.

[GKP88] G. Georgakopoulos, D. Kavvadias, and C. H. Papadimitriou. Probabilistic satisfiability. *J. of Complexity*, 4(1):1–11, 1988.

[HJ00] P. Hansen and B. Jaumard. Probabilistic satisfiability. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems, vol.5.* Springer, 2000.

[JFSS06] B. Jaumard, A. Fortin, I. Shahriar, and R Sultana. First order probabilistic logic. In *NAFIPS 2006*, pages 341–346, 2006.

[KP90] D. Kavvadias and C. H. Papadimitriou. A linear programming approach to reasoning about probabilities. *AMAI*, 1:189–205, 1990.

[LS10] Carsten Lutz and Lutz Schröder. Probabilistic description logics for subjective uncertainty. In *KR 2010, 12th International Conference of Knowledge Representation and Reasoning*. AAAI Press, 2010.

[Luk08] T Lukasiewicz. Expressive probabilistic description logics. *Artificial Intelligence*, 172(6-7):852–883, 2008.

[GM10] O. Gries, and R. Möller. Gibbs sampling in probabilistic description logics with deterministic dependencies. *First International Workshop on Uncertainty in Description Logics (Uni-DL)*, 2010.

[CL06] P. C. G. Costa and K. B. Laskey. PR-OWL: A framework for probabilistic ontologies. *Conf. on Formal Ontology in Information Systems*, 2006.

[DFL08] C. D'Amato, N. Fanizzi, and T. Lukasiewicz. Tractable reasoning with Bayesian description logics. *Int. Conf. on Scalable Uncertainty Management* (LNAI 5291), pages 146–159, 2008.

[DPP06] Z. Ding, Y. Peng, and R. Pan. BayesOWL: Uncertainty modeling in semantic web ontologies. *Soft Computing in Ontologies and Semantic Web*, pages 3–29. Springer, Berlin/Heidelberg, 2006.

[DS05] M. Dürig and T. Studer. Probabilistic ABox reasoning: preliminary results. *Description Logics*,pages104–111,2005.

[GT07] L. Getoor and B. Taskar. *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[Hal03] J. Y. Halpern. *Reasoning about Uncertainty.* MIT Press, Cambridge, Massachusetts, 2003.

[Hei94] J. Heinsohn. Probabilistic description logics. *Conf. Uncertainty in AI*, page 311-318, 1994.

[Jae94] M. Jaeger. Probabilistic reasoning in terminological logics. *Principles of Knowledge Representation*, pages 461–472, 1994.

[KLP97] D. Koller, A. Y. Levy, and A. Pfeffer. P-CLASSIC: A tractable probablistic description logic. *AAAI Conf. on AI*, pages 390–397, 1997.

[LS08] T. Lukasiewicz and U. Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *Journal of Web Semantics*, 6(4):291–308, November 2008.

[Seb94] F. Sebastiani. A probabilistic terminological logic for modelling information retrieval. *17th Conf. on Research and Development in Information Retrieval*, pages 122–130, Dublin, Ireland, 1994. Springer-Verlag.