University of São Paulo
Institute of Mathematics and Statistics
Bachelor of Computer Science

Tractable Probabilistic Description Logic

Algorithms and Implementation

Andrew Ijano Lopes

Final Essay

mac 499 — Capstone Project

Program: Computer Science

Advisor: Prof. Dr. Marcelo Finger

São Paulo
January 20th, 2021

Tractable Probabilistic Description Logic

Algorithms and Implementation

Andrew Ijano Lopes

This is the original version of the
capstone project report prepared by
the candidate Andrew Ijano Lopes, as
submitted to the Examining Committee.

Resumo

Andrew Ijano Lopes. **Lógica de Descrição Probabilística Tratável: *Algoritmos e Implementação***. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2020.

Elemento obrigatório, constituído de uma sequência de frases concisas e objetivas, em forma de texto. Deve apresentar os objetivos, métodos empregados, resultados e conclusões. O resumo deve ser redigido em parágrafo único, conter no máximo 500 palavras e ser seguido dos termos representativos do conteúdo do trabalho (palavras-chave). Deve ser precedido da referência do documento.

Palavras-chave: Lógicas de descrição. Palavra-chave2. Palavra-chave3.

Abstract

Andrew Ijano Lopes. **Tractable Probabilistic Description Logic: *Algorithms and Implementation***. Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2020.

Elemento obrigatório, elaborado com as mesmas características do resumo em língua portuguesa. De acordo com o Regimento da Pós-Graduação da USP (Artigo 99), deve ser redigido em inglês para fins de divulgação. É uma boa ideia usar o sítio www.grammarly.com na preparação de textos em inglês.

Keywords: Description logics. Keyword2. Keyword3.

Lista de Abreviaturas

DL Description Logic

Lista de Símbolos

List of Figures

List of Tables

List of Programs

Contents

Todo list

Chapter 1

Introduction

Need to rewrite, but the example is OK

mfinger: meus comentários em verde

Description logics are a family of formal knowledge representation languages, being of particular importance in providing a logical formalism for ontologies and the Semantic Web. Also, they are notable in biomedical informatics for assisting the codification of biomedical knowledge. Due to these uses, there is a great demand to find tractable (i.e., polynomial-time decidable) description logics.

One of them, the logic $\mathcal{EL}^{++}$, is one of the most expressive description logics in which the complexity of inferential reasoning is tractable (Baader *et al.*, 2005). Even though it is expressive enough to deal with several practical applications, there was also a need to model situations in which a General Concept Inclusion Axiom is not always true, which has already been proposed in the literature (Boole, 1854).

Example 1.1

> Consider a medical situation, in which a patient may have non-specific symptoms, such as high fever, cough, and headache. Also, COVID-19, a severe acute respiratory syndrome caused by the SARS-CoV-2 virus, is a disease that can account for those symptoms, but not all patients present all symptoms. Such an uncertain situation is suitable for probabilistic modeling.
>
> In a certain hospital, a patient with a high fever has some probability of having COVID-19, but that probability is 20% larger if the patient has a cough too. On the other hand, COVID-19 is not very prevalent and is not observed in the hospital 90% of the time. If those probabilistic constraints are satisfiable, one can also ask the minimum and maximum probability that a hospital patient Mary, with fever and cough, is a suspect of suffering from COVID-19.

For classical propositional formulas, this problem, called *probabilistic satisfiability* (PSAT), has already been presented with tractable fragments (Andersen and Pretolani, 2001). On the other hand, in description logics, most studies result in intractable reasoning;

moreover, by adding probabilistic reasoning capabilities to $\mathcal{EL}^{++}$, in order to model such situation, the complexity reaches NP-completeness (Finger, 2019).

To solve this problem, probabilistic constrains can be applied to axioms and its probabilistic satisfaction can be seen in a linear algebraic view. Furthermore, it can be reduced to an optimization problem, which can be solved by an adaptation of the simplex method with column generation. (Finger, 2019) Thus, it is possible to reduce the column generation problem to the *weighted partial maximum satisfatibility*.

Then, recent studies show that it is necessary to focus on a fragment of $\mathcal{EL}^{++}$ for obtain a tractable probabilistic reasoning. This fragment is called *Graphic* $\mathcal{EL}^{++}$ ($\mathcal{GEL}$) and it is defined as an $\mathcal{EL}^{++}$-fragment in which its set of axioms and *role inclusions* contains formulas in *normal form* and does not allow explicit conjunction axioms. Therefore, axioms can be seen as edges in a graph, as opposed to hyperedges in a hypergraph, which is the case of $\mathcal{EL}^{++}$. This allows the use of graph-based machinery to develop tractable algorithm for the *weighted partial Maximum SATisfatibility* for $\mathcal{GEL}$ (Max $\mathcal{GEL}$-SAT) and, as a result, a tractable probabilistic description logic.

1.1 Objectives

- Investigate a potentially tractable fragment of $\mathcal{EL}^{++}$;
- Study and implement tractable algorithms for the problem of *weighted partial* Max $\mathcal{GEL}$-SAT;
- Study and implement algorithms for the problem of *probabilistic satisfatibility for* $\mathcal{GEL}$ (PGEL-SAT), using the Max $\mathcal{GEL}$-SAT solver as a subroutine. Thus, it is excepted to achieve a tractable algorithm for a probabilistic description logic.

1.2 Structure of this work

In this paper, we describe the implementation of these algorithms [1] and is organized as follows: Section **??** highlights related results in the literature. The basic definition of $\mathcal{GEL}$ with its algorithms for MaxSAT and PSAT are described in Section **??** and followed by Section **??**, which presents details about the implementation and its experimental evaluation.

[1] Available at https://github.com/AndrewIjano/pgel-sat

Chapter 2

Background

In this chapter, we present the theoretical background of description logics,

Add initial description of the chapter

2.1 Description logics

Description logics (DLs) are used to represent knowledge, such as the semantic of words, people and their relations, and medical terms. These scenarios require precise specification and meaning so that different systems behave the same way. The first DL modeling languages appeared in the mid-1980s and have an important role in the context of the Semantic Web, an initiative to represent web content in a form that is more machine friendly (Krötzsch *et al.*, 2012).

As their name suggests, DLs are logics; indeed, most of them are fragments of first-order logic. This relation with logics is what provides their precise specification, called *formal semantics*. Also, it equips their languages with a formal deduction to *infer* additional information, and the computation of these inferences is called *reasoning*. The performance of algorithms for reasoning strongly relies on the expressiveness of the DL: fast algorithms usually exist for lightweight logics. Then, there is not just a single DL because the balance between expressiveness and performance depends on the application. (Krötzsch *et al.*, 2012)

2.1.1 Building blocks of description logic ontologies

A DL is composed of concepts, roles, and individual names. Concepts are sets of individuals, roles are binary relations between individuals and individual names are single individuals in the domain.

For example, an ontology modeling the situation in Example 1.1 can use the concepts Patient, to represent the set of all patients in the hospital, and Symptom, to represent the set of all symptoms; roles such hasSymptom, to represent the binary relation between

Maybe explain what is an ontology

patients and symptoms; and individual names such as mary and s1, to represent the individuals Mary and Mary's symptoms.

Additionally, DLs allows us to describe more complex situations, creating new concepts and roles from the previously defined ones.

Some concept constructors provide boolean operations similar to that found in set theory and logic expressions. For example, if we want to describe the set of individuals that are both fever and cough, we could use the *conjunction* operator, as follows

$$\mathsf{Fever} \sqcap \mathsf{Cough}.$$

We can link concepts and roles using role restrictions. For example, to describe all individuals that are suspect of some disease that is COVID-19, we use the *existential restriction*

$$\exists \mathsf{suspectOf}.\mathsf{COVID\text{-}19}.$$

Also, to define concepts with only one individual we use *nominals* like $\{\mathsf{mary}\}$.

More expressive logics can have other operations such as *disjunction* ($C \sqcup D$), *negation* ($\neg C$), *universal restriction* ($\forall r.C$) and *number restrictions* ($\leq n\, r.C$).

To capture knowledge about the world, DL ontologies also allow us to describe relations between concepts, roles, and individual names. For example, the fact that all fevers are symptoms is represented by the *concept inclusion*

$$\mathsf{Fever} \sqsubseteq \mathsf{Symptom};$$

the knowledge that someone that has a symptom which is caused by some disease is suspect of that disease can be expressed by the *role inclusion* with a *role composition*

$$\mathsf{hasSymptom} \circ \mathsf{hasCause} \sqsubseteq \mathsf{suspectOf};$$

and the fact that Mary is a patient of the hospital and has symptoms is represented by the *assertions* Patient(mary) and hasSymptom(mary, s1).

After that, if we have a set of these relations, one could ask if there is a set of individuals, or instances, that satisfies these relations, which is called an *interpretation*. Interpretations can be understood as the assignment of meaning to logical terms in an ontology. Because a DL usually considers all the possibles situations, property that is sometimes referred to as *open world assumption*, an ontology can have multiple satisfiable interpretations. The fewer restrictions it has, the more interpretations satisfy this ontology. The computational complexity to find the existence of these interpretations is one of the key aspects to choose different DL fragments.

These terms will be formally defined in the section 2.2, in the case of the DL $\mathcal{EL}^{++}$.

2.1.2 Description logic fragments and OWL

There are many DL fragments. Each subset of features, like those described previously, can lead to different fragments of first-order logic. For example, the logic $\mathcal{ALC}$ does not allow role inclusions and admits only $\sqcap, \sqcup, \neg, \exists$ and $\forall$ as concept constructors; their best reasoning algorithms, however, are worst-case exponential time. On the other hand, the $\mathcal{EL}$ logic allows only $\sqcap$ and $\exists$ as concept constructors, and its reasoning algorithms are polynomial time.

To express DL ontologies, the World Wide Web Consortium (W3C) designed the OWL 2 Web Ontology Language (OWL 2) (Hitzler *et al.*, 2009). This declarative language is part of the W3C's Semantic Web technology stack and comes with various syntaxes, such as RDF/XML. Because of this use on the web, names in OWL are *international resource identifiers* (IRIs).

2.2 The description logic $\mathcal{EL}^{++}$

$\mathcal{EL}^{++}$ is an extension of the DL $\mathcal{EL}$ (Baader *et al.*, 2005). It was created with large bio-health ontologies in mind, such as SNOMED-CT, the NCI thesaurus, and Galen, and became an official OWL 2 profile (Hitzler *et al.*, 2009). We concentrate on presenting $\mathcal{EL}^{++}$ without concrete domains.

2.2.1 Syntax

In $\mathcal{EL}^{++}$, *concept descriptions* are defined inductively from a set $\mathsf{N_C}$ of *concept names*, a set $\mathsf{N_R}$ of *role names* and set $\mathsf{N_I}$ of *individual names* as follows:

- $\top$, $\bot$ and concept names in $\mathsf{N_C}$ are concept descriptions;
- if C and D are concept descriptions, $C \sqcap D$ is a concept description;
- if C is a concept description and $r \in \mathsf{N_R}$, $\exists r.C$ is a concept description;
- if $a \in \mathsf{N_I}$, $\{a\}$ is a concept description.

To represent knowledge using concept descriptions, we need to define facts (axioms and role inclusions) and assertions.

An *axiom*, or a *general concept inclusion* (GCI), is an expression of the form $C \sqsubseteq D$, where C and D are concept inclusions. Also, we write $C \equiv D$ to represent the axioms $C \sqsubseteq D$ and $D \sqsubseteq C$. A *role inclusion* (RI) is an expression of the form $r_1 \circ \cdots \circ r_k \sqsubseteq r$, where $r_1, \ldots r_k, r \in \mathsf{N_R}$. The symbol "$\circ$" denotes composition of binary relations. A *constraint box* (CBox) is a finite set of GCIs and a finite set of RIs.

Similarly, a *concept assertion* is an expression of the form $C(a)$ and a *role assertion*, $r(a, b)$, where C is a concept description, $a, b \in \mathsf{N_I}$ and $r \in \mathsf{N_R}$. A finite set of concept assertions and role assertions is an *assertional box* (ABox).

Then, an $\mathcal{EL}^{++}$ *knowledge base* $\mathcal{K}$ (KB) is a pair $(\mathcal{C}, \mathcal{A})$, where $\mathcal{C}$ is a CBox and $\mathcal{A}$ is an ABox.

2.2.2 Semantics

The semantics of $\mathcal{EL}^{++}$ are given by *interpretations* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. The *domain* $\Delta^{\mathcal{I}}$ is a non-empty set of individuals, and the *interpretation function* $\cdot^{\mathcal{I}}$ maps each concept name $A \in \mathsf{N_C}$ to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, each role name r to a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and each individual name a to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The extension of $\cdot^{\mathcal{I}}$ for an arbitrary concept description is inductively defined by the third column of Table 2.1.

Name	Syntax	Semantics
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} \ : \ (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
GCI	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
RI	$r_1 \circ \cdots \circ r_k \sqsubseteq r$	$r_1^{\mathcal{I}} \circ \cdots \circ r_k^{\mathcal{I}} \subseteq r^{\mathcal{I}}$
concept assertion	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$r(a, b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$

Table 2.1: *Syntax and semantics of $\mathcal{EL}^{++}$ without concrete domains*

The interpretation $\mathcal{I}$ *satisfies*:

- an axiom $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ (represented as $I \models C \sqsubseteq D$);
- a RI $r_1 \circ \cdots \circ r_k \sqsubseteq r$ if $r_1^{\mathcal{I}} \circ \cdots \circ r_k^{\mathcal{I}} \subseteq r^{\mathcal{I}}$ (represented as $I \models r_1 \circ \cdots \circ r_k \sqsubseteq r$);
- an assertion $C(a)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ (represented as $I \models C(a)$);
- an assertion $r(a, b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$ (represented as $I \models r(a, b)$).

Also, we say that $\mathcal{I}$ is a *model* of:

- a CBox $\mathcal{C}$ if it satisfies every axiom and RI in $\mathcal{C}$ (represented as $\mathcal{I} \models \mathcal{C}$);
- an ABox $\mathcal{A}$ if it satisfies every assertion in $\mathcal{A}$ (represented as $\mathcal{I} \models \mathcal{A}$);

Then, an important problem in $\mathcal{EL}^{++}$ is to determine its *consistency*, that is if $\mathcal{A}$ and $\mathcal{C}$ have a common model, which is in PTime (Baader *et al.*, 2005).

2.2.3 Normal form

We can convert an $\mathcal{EL}^{++}$ knowledge base into a normal form, in polynomial time, by introducing new concept and role names (Baader *et al.*, 2005).

First, there is no need of explicit ABox, because $\mathcal{I} \models C(a) \iff \mathcal{I} \models \{a\} \sqsubseteq C$ and $\mathcal{I} \models r(a, b) \iff \{a\} \sqsubseteq \exists r.\{b\}$. In other words, a knowledge base can be represented by just a CBox, by transforming assertions in axioms.

In addition, given a CBox $\mathcal{C}$, consider the set $\mathsf{BC}_\mathcal{C}$ of *basic concept descriptions*, which is the smallest set of concept descriptions that contains the top concept $\top$, all concept names used in $\mathcal{C}$ and all concepts of the form $\{a\}$ used in $\mathcal{C}$.

Then, every axiom can be represented in the following normal form, where $C_1, C_2 \in \mathsf{BC}_\mathcal{C}$, $D \in \mathsf{BC}_\mathcal{C} \cup \{\bot\}$:

$$C_1 \sqsubseteq D \qquad \text{(simple)}$$
$$C_1 \sqsubseteq \exists r.C_2 \qquad \text{(existential-head)}$$
$$\exists r.C_1 \sqsubseteq D \qquad \text{(existential-body)}$$
$$C_1 \sqcap C_2 \sqsubseteq D \qquad \text{(conjunctive-body)}$$

And every RI are of the form $r \sqsubseteq s$ or $r_1 \circ r_2 \sqsubseteq s$.

Example 2.1

Consider the following CBox $\mathcal{C}_{exa}$ representing the situation in Example 1.1. On the left, we have basic knowledge of diseases and, on the right, the specific knowledge about Mary. Note that, for simplicity, it is not in normal form.

Fever ⊑ Symptom	
Cough ⊑ Symptom	
COVID-19 ⊑ Disease	{mary} ⊑ Patient
Symptom ⊑ ∃hasCause.Disease	{s1} ⊑ Fever ⊓ Cough
Patient ⊑ ∃hasSymptom.Symptom	{mary} ⊑ ∃hasSymptom.{s1}
CovidPatient ≡ ∃suspectOf.COVID-19	
hasSymptom ∘ hasCause ⊑ suspectOf	

Because CBoxes can only represent facts, there is no way to describe uncertain knowledge. Even though, in cases when which of them are true, we could define three axioms

Ax_1 := Fever ⊑ ∃hasCause.COVID-19, when fever is actually caused by COVID-19;

Ax_2 := Fever ⊓ Cough ⊑ ∃hasCause.COVID-19 , when both fever and cough are caused by COVID-19;

Ax_3 := COVID-19 ⊑ ⊥, when there are no presence of COVID-19 in the hospital.

In the following sections, it will be presented how to add these axioms in a KB with probabilistic properties.

We want to model uncertain information using DLs. However, it has been proved that, by adding probabilistic reasoning capabilities to $\mathcal{EL}^{++}$, the complexity reaches NP-completeness (Finger, 2019). Then, it is necessary to reduce the expressiveness of this language.

2.3 Graphic $\mathcal{EL}$ ($\mathcal{GEL}$)

Graphic $\mathcal{EL}$ ($\mathcal{GEL}$) is a fragment of $\mathcal{EL}^{++}$ in which every axiom and RI are in normal form and no conjunctive-body axiom is allowed (Finger, n.d.). The semantics are the same as that of $\mathcal{EL}^{++}$.

Example 2.2

> Since there are conjunctive-body axioms in the CBox in Example 2.1, we need to modify this knowledge in order to represent it in $\mathcal{GEL}$. First, we substitute every concept description Fever ⊓ Cough by a new basic concept FeverAndCough. After that, we add axioms FeverAndCough ⊑ Fever and FeverAndCough ⊑ Cough to CBox.

The name of this fragment comes from the fact that each GCI can be represented as arrows in a graph where nodes are basic concepts. This representation is useful for the development of algorithms and it is used to define its SAT decision.

2.3.1 Graphical representation

Consider a $\mathcal{GEL}$-CBox $\mathcal{C}$ with n_R roles, its graphical representation is a edge-labeled graph $G(\mathcal{C}) := (N, E, \ell)$, where N is a set of nodes, $E \subseteq N^2$ is a set of directed edges and $\ell : E \longrightarrow \{0, 1, \dots, n_R\}$ is a labeling function.

How do I define ℓ with the addition of artificial edges of the form $u_{r.C}$?

In addition, Finger (n.d.) defines the following notation. The set $E_i \subseteq E$ is the set of all edges e such that $\ell(e) = i$. We write $X_1 \longrightarrow_i X_2$ if there is an edge $(X_1, X_2) \in E_i$, and $X_1 \not\longrightarrow_i X_2$ if $(X_1, X_2) \notin E_i$. The expression $X \longrightarrow_i^* Y$ represents the reflexive transitive closure of $\longrightarrow_i$, which is the existence of a path in the graph of size ≥ 0, starting in X, ending in Y, and only going through edges in E_i, for $0 \leq i \leq n_R$. Finally, $X \rightsquigarrow Y$ represents a path from X to Y using any type of edge.

Then, the graph $G(\mathcal{C}) = (N, E, \ell)$ can be constructed from a CBox $\mathcal{C}$ with the following steps:

1. for each concept $\exists r.C$ in a existential-body axiom of the form $\exists r.C \sqsubseteq D$ create a *virtual concept* "$\exists r.C$". The set of all virtual concepts in $\mathcal{C}$ is called $\mathsf{VC}_\mathcal{C}$;
2. the set N of nodes is obtained from the basic concepts of $\mathcal{C}$, an initial-node symbol $Init$, the bottom concept $\bot$ and the set of virtual concepts of $\mathcal{C}$ as follows:

$$N := \{Init, \bot\} \cup \mathsf{BC}_\mathcal{C} \cup \mathsf{VC}_\mathcal{C};$$

3. if $C \sqsubseteq D \in \mathcal{C}$ then $C \longrightarrow_0 D$;

In the last paragraph, I wrote "We write $X_1 \longrightarrow_i X_2$ if there is an edge $(X_1, X_2) \in E_i$". So the right notation is $C \longrightarrow_0 D$ instead of $C \longrightarrow_0 D \in E_0$ (the one you wrote in the paper), isnt it? Doesnt $X_1 \longrightarrow_i X_2$ already mean the existence of the edge in E?

4. if $C \sqsubseteq \exists r_i.D$ then $C \longrightarrow_i D$;
5. if $\exists r_i.C \sqsubseteq D$ then "$\exists r_i.C$" $\longrightarrow_0 D$, "$\exists r_i.C$" $\longrightarrow_i C$ and $C \longrightarrow_{u_{r_i.C}}$ "$\exists r_i.C$";
6. $Init \longrightarrow_0 \top$;

Need to review this. I dont think the theory is finalized

The original rule was "if $\top$ occurs in C then $Init \longrightarrow_0 \top$". Can I change this?

7. for every node of the form $\{a_i\} \in N$, $Init \longrightarrow_0 \{a_i\}$.

Example 2.3

Consider the CBox in Example 2.2 and the uncertain information in Example 2.1. Its graphical representation is displayed in the Figure 2.1. The $\longrightarrow_0$-edges are represented by continuous black arrows, the $\longrightarrow_i$-edges, $i \geq 1$, are represented by blue labeled arrows and red dotted arrows indicate that source is uncertain information.

Note that the axiom CovidPatient $\equiv \exists$suspectOf.COVID-19 implies the existential-body axiom $\exists$suspectOf.COVID-19 $\sqsubseteq$ CovidPatient. Therefore, it generates:

- the virtual concept "suspectOf.COVID-19";
- "suspectOf.COVID-19" $\longrightarrow$ CovidPatient;
- "suspectOf.COVID-19" $\longrightarrow_{\text{suspectOf}}$ CovidPatient;
- and COVID-19 $\longrightarrow_{u_{\text{suspectOf.Covid}}}$ "suspectOf.COVID-19".

Is this the right notation? Doesn't the label need to be an integer?

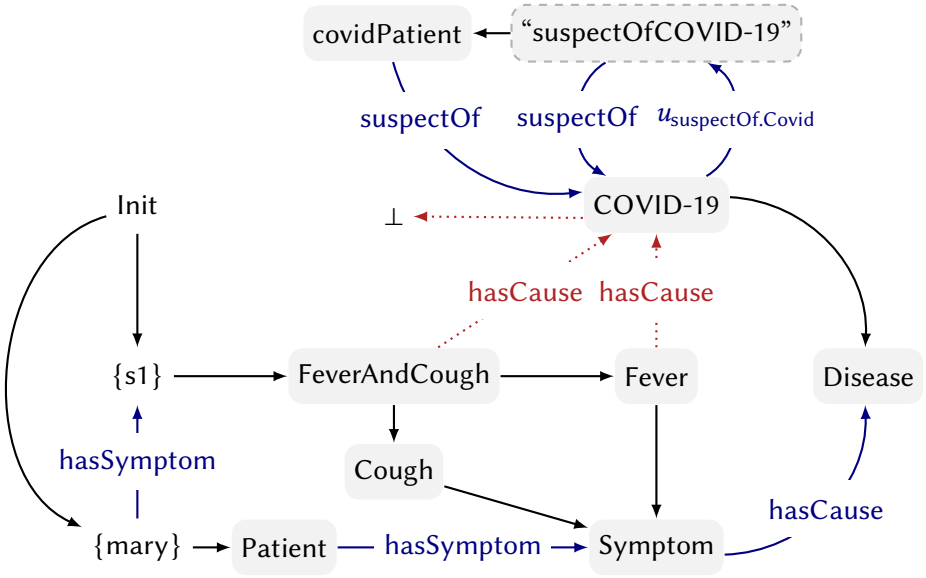

Figure 2.1: *Graphical representation of the ontology in Example 2.2.*

2.3.2 SAT decision

Need to describe the graph completion rules

Question about this part: do we really need the graph completion to define the SAT decision? Isn't it true that a $\mathcal{GEL}$ CBox $\mathcal{C}$ is **unsatisfiable** iff $Init \rightsquigarrow \bot$ (using any type of edge) in a graph $G(\mathcal{C})$ **without** completion? I couldn't find any counterexample...

After the graph completion, the satisfiability (SAT) decision of a $\mathcal{GEL}$ CBox can be reduced to a path search in a graph, that is, a $\mathcal{GEL}$ CBox is unsatisfiable iff $Init \longrightarrow_0^* \bot$.

2.4 MaxSAT for $\mathcal{GEL}$

Before focusing on the probabilistic extension for $\mathcal{GEL}$, we need to define its MaxSAT problem, which will compose further the probabilistic reasoner.

The *weighted partial maximum satisfiability problem for* $\mathcal{GEL}$ ($\mathcal{GEL}$-MaxSAT) can be defined as follows: given a potentially inconsistent weighted CBox $\mathcal{C}$, we want to find the maximal satisfiable subset of axioms; since it is partial, some axioms must be present in this subset. Usually, partiality can be modeled assigning infinite weights to the axioms that must not be excluded.

A *weighted* CBox is a pair $\langle \mathcal{C}, w \rangle$ where $\mathcal{C}$ is a CBox and $w : \mathcal{C} \longrightarrow \mathbb{Q} \cup \{\infty\}$ is a weight function, which maps axioms in $\mathcal{C}$ to weights. The infinite weight is used to represent axioms that must not be excluded in the maximal satisfiable subset. Also, it is defined that RIs of the form $r \sqsubseteq s$ and $r_1 \circ r_2 \sqsubseteq s$ always have infinite weight.

Then, given a weighted CBox $\langle \mathcal{C}, w \rangle$, a solution for the weighted partial $\mathcal{GEL}$-MaxSAT problem is a set $\mathcal{C}_{max} \subseteq \mathcal{C}$ such that:

- $\mathcal{C}_{max}$ is satisfiable; and
- $\mathcal{C}_{max} \models C \sqsubseteq D$ if $w(C \sqsubseteq D) = \infty$; and
- the sum of finite weights in $\mathcal{C}_{max}$ is maximal.

2.5 Probabilistic $\mathcal{GEL}$

Probability in $\mathcal{GEL}$ is constructed from a *probability function* P (Finger, n.d.). Consider a finite number of interpretation, $\mathcal{I}_1, \dots, \mathcal{I}_m$, we define the probability function $P : \{\mathcal{I}_1, \dots, \mathcal{I}_m\} \longrightarrow \mathbb{Q}$, such that $P(\mathcal{I}_i) \geq 0$ and $\sum_{i=1}^{m} P(\mathcal{I}_i) = 1$. We can also define the probability of an axiom $C \sqsubseteq D$ as follows

$$P(C \sqsubseteq D) = \sum_{\mathcal{I}_i \models C \sqsubseteq D} P(\mathcal{I}_i).$$

A *probabilistic knowledge base* is a pair $\langle \mathcal{C}, \mathcal{P} \rangle$, where $\mathcal{C}$ is a CBox and $\mathcal{P}$ is a PBox. A PBox is a set of k linear constraints over n axioms, of the form

$$\sum_{j=1}^{n} a_{ij} \cdot P(C_j \sqsubseteq D_j) \leq b_i; \quad 1 \leq i \leq k. \tag{2.1}$$

We can define the *satisfiability problem* for this probabilistic KB (PGEL-SAT) as deciding if it is consistent or not. If it is consistent, the solution is a set of interpretations $\{\mathcal{I}_1, \dots, \mathcal{I}_m\}$ and a probability function $P : \{\mathcal{I}_1, \dots, \mathcal{I}_m\} \longrightarrow \mathbb{Q}^+$ such that $\sum_{i=1}^{m} P(\mathcal{I}_i) = 1$, $P(C \sqsubseteq D) = 1$ for $C \sqsubseteq D \in \mathcal{C}$ (axioms in CBox are certain) and P verifies all linear constraints in $\mathcal{P}$.

Example 2.4

Now we can model the uncertain situation stated in Example 1.1 using the probability knowledge base $\langle \mathcal{C}_{exa}, \mathcal{P}_{exa} \rangle$, where $\mathcal{C}_{exa}$ is the CBox from Example 2.2 and $\mathcal{P}_{exa}$ is given by

$$\mathcal{P}_{exa} := \{ \quad P(Ax_2) - P(Ax_1) = 0.2, \quad P(Ax_3) = 0.9 \quad \}.$$

Then, we need a polynomial algorithm to find if this probabilistic KB is consistent.

2.5.1 Linear algebraic view

The PGEL-SAT problem was also defined by Finger (n.d.) in a linear algebraic view, which is useful to develop its polynomial reasoning algorithm. It was shown that a probabilistic KB $\langle \mathcal{C}, \mathcal{P} \rangle$ is satisfiable iff the linear equation $C \cdot x = d$ has a solution $x \geq 0$, where

$$C := \begin{bmatrix} -I_n & M_{n\times m} \\ A_{k\times n} & 0_{k\times m} \\ 0'_n & 1'_m \end{bmatrix} \quad x := \begin{bmatrix} p_n \\ \pi_m \end{bmatrix} \quad d := \begin{bmatrix} 0_n \\ b_k \\ 1 \end{bmatrix} \tag{2.2}$$

and

- $A_{k\times n}$ is a $k \times n$ matrix whose elements a_{ij} are given by Equation 2.1;
- b_k is a k vector whose elements b_i are also given by Equation 2.1;
- $M_{n\times m}$ is a $n \times m$ matrix given by the following steps:

  Consider an interpretation $\mathcal{I}$ model of $\mathcal{C}$, its corresponding vector in $\mathcal{P}$ is a $\{0,1\}$-vector y such that $y_i = 1$ iff $\mathcal{I} \models C_i \sqsubseteq D_i$, for $1 \leq i \leq n$.

  Then, given a set of interpretations $\mathcal{I}_1, \ldots, \mathcal{I}_m$, we define $M_{n\times m}$ a matrix whose column M^j is $\mathcal{I}_j$'s corresponding vector in $\mathcal{P}$;
- I_n is the n-dimensional identity matrix;
- 0_n is a column 0-vector of size n (similarly for 1_n);
- $0'_n$ is the previous vector's transpose;
- $0_{k\times m}$ is a 0-matrix of shape $k \times m$;
- p_n is a vector of size n which corresponds to the probability of axioms occurring in Equation 2.1;
- π_m is a vector of size m which corresponds to the probability distribution over interpretations $\mathcal{I}_1, \ldots, \mathcal{I}_m$.

Thus, we can use techniques for solving linear equations to find a tractable algorithm for PGEL-SAT.

Chapter 3

Development

In this section, we describe the development of a tractable algorithm for PGEL-SAT[1], described by Finger (n.d.). It was implemented using Python programming language (Van Rossum and Drake, 2009).

3.1 Input and output format

The algorithm accepts as input a P$\mathcal{GEL}$ KB encoded in an OWL 2 ontology. Both certain and uncertain knowledge must be a $\mathcal{GEL}$-CBox in the normal form, with the additional support of equivalence axioms. Due to limitations in the OWL parser used, which will be detailed further, RDF/XML, OWL/XML and NTriples are the only file formats supported.

In addition, uncertain axioms must have an annotation (`rdfs:comment`) with its unique numerical index. That is, given an uncertain axiom Ax_i, its annotation must be of the following form in Figure 3.1.

```
#!pbox-id i
```

Figure 3.1: *Annotation format for an uncertain axiom*

PBox restrictions are represented with annotations in the $\top$ concept (`owl:Thing`). That is, given a restriction of the form $a_0P(Ax_0) + a_1P(Ax_1) + \cdots + a_{n-1}P(Ax_{n-1}) = b$, its annotation must be of the form in Figure 3.2. Also, inequalities $\leq$ and $\geq$ can be represented, respectively, with the symbols <= and >=.

The output of the algorithm is `True` if the given KB has a solution, and `False` if not.

Maybe explain how to execute the algorithm

[1] Available at https://github.com/AndrewIjano/pgel-sat.

```
#!pbox-restriction
0 a_0
1 a_1
...
n - 1 a_{n-1}
==
b
```

Figure 3.2: *Annotation format for a PBox restriction*

3.2 Knowledge base representation

To read the ontology in OWL 2, it was used the Python module *Owlready2* (Lamy, 2017).

The probabilistic KB is represented as the edge-labeled graph in subsection 2.3.1 with three matrices for the inequalities in Equation 2.1.

Need to improve this

The probabilistic KB representation allows us to develop a PGEL-SAT solver, using this data structure as input.

3.3 PGEL-SAT solver

In order to understand the PGEL-SAT solver implemented, we need to find a solution for the linear system stated in subsection 2.5.1.

First, consider the matrix C in Equation 2.1, it has p-columns and π-columns, referring to which part of x they multiply. We say that a p-column pc_i is *well-formed* if it is of the form in Equation 2.1, starting with the i-th column of $-I_n$, followed by the i-th column of A and with one 0 at the end. Also, we say that a π-column is *well-formed* if it starts with a corresponding vector of an interpretation model of $\mathcal{C}$, followed by k zeros and with one 1 at the end. A column that is not well-formed is *ill-formed*.

Now, consider that the matrix C may have ill-formed columns. We define a binary *cost vector* c such that each element $c_i = 1$ iff column C^i is ill-formed; otherwise $c_i = 0$. Then, the linear system stated in subsection 2.5.1 has a solution iff the following minimization problem has minimum 0.

$$\begin{aligned} \text{minimize} \quad & c' \cdot x \\ \text{subject to} \quad & C \cdot x = d \\ & x \geq 0 \end{aligned}$$

This problem can be solved by a linear algebraic solver but we need to find beforehand C and d that reaches the minimum 0. Since matrix A and vector b are generated from

PBox, we need to choose a set of interpretations that provides the solution. However, we potentially have an exponential collection of possible interpretations, and looking over all of them would make the algorithm untractable.

Need to say about M being large not the collection of possible columns

Even though, from Carathéodory's Theorem (Eckhoff, 1993), Finger (n.d.) states that if constrains in Equation 2.1 are solvable then there exists a solution where x has at most $n + k + 1$ values such that $x_j > 0$.

- there are many possible columns

- need to generate the right ones

- column generation + linear solver

- columns are generated on the fly

- the start

- explain the program

- the program

3.4 Linear solver

- explain the implementation; lib used: glpk python

- interior points -> tractable

3.5 Column generation

- explain the theory of the solution

- pseudocode of the algorithm

3.6 GEL-MaxSAT

- explain the theory of the solution

- pseudocode of the algorithm

- max flow / min cut

Chapter 4

Experiments

Chapter 5

Results

Chapter 6

Related work

The problem of probabilistic reasoning and extensions in logics to deal with uncertainty have been studied for several decades. The first known proposal of PSAT, for propositional formulas, is attributed to Boole (1854) and it has already been shown to be NP-Complete (Georgakopoulos *et al.*, 1988).

In the relational domain, the literature contain several logics with probabilistic reasoning capabilities although they have led to intractable decision problems. Some of them extend the already intractable $\mathcal{ALC}$, with probabilistic constrains over concepts (Heinsohn, 1994; Lukasiewicz, 2008; Gutiérrez-Basulto *et al.*, 2011). For the expressive and lightweight $\mathcal{EL}$-family, some extensions such as Gutiérrez-Basulto *et al.* (2017) and Ceylan and Peñaloza (2017) have led to ExpTime-hard or PP-complete probabilistic reasoning; futhermore, NP-completeness can be achieved with probability capabilities over axioms (Finger, 2019).

On the other hand, many results implies that the research on Max-SAT has a impact on the solutions of PSAT problems (Andersen and Pretolani, 2001). Also, there was already proposed a MaxSAT-solver for a propositional fragment of horn logic by a max-flow/min-cut formulation (Jaumard and Simeone, 1987). Thus, it is expected to ask if one could also take such results to a relational domain.

Chapter 7

Conclusion and future work

References

[Andersen and Pretolani 2001] Kim Allan Andersen and Daniele Pretolani. "Easy cases of probabilistic satisfiability". In: *Annals of Mathematics and Artificial Intelligence* 33.1 (2001), pp. 69–91 (cit. on pp. 1, 21).

[Baader *et al.* 2005] Franz Baader, Sebastian Brandt, and Carsten Lutz. "Pushing the EL Envelope". In: *Proceedings of the 19th International Joint Conference on Artificial Intelligence*. IJCAI'05. 2005, pp. 364–369 (cit. on pp. 1, 5, 6).

[Boole 1854] G. Boole. *An Investigation of the Laws of Thought: On which are Founded the Mathematical Theories of Logic and Probabilities*. Collected logical works. Walton and Maberly, 1854. URL: https://books.google.com.br/books?id=DqwAAAAAcAAJ (cit. on pp. 1, 21).

[Ceylan and Peñaloza 2017] Ismail Ilkan Ceylan and Rafael Peñaloza. "The Bayesian Ontology Language BEL". In: *Journal of Automated Reasoning* 58.1 (2017), pp. 67–95 (cit. on p. 21).

[Eckhoff 1993] Jürgen Eckhoff. "Helly, Radon, and Carathéodory type theorems". In: *Handbook of convex geometry*. Elsevier, 1993, pp. 389–448 (cit. on p. 15).

[Finger 2019] Marcelo Finger. "Extending EL++ with Linear Constraints on the Probability of Axioms". In: *Description Logic, Theory Combination, and All That. Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday*. Ed. by Carsten Lutz, Uli Sattler, Cesare Tinelli, Anni-Yasmin Turhan, and Frank Wolter. Vol. LLNCS 11560. Theoretical Computer Science and General Issues. Springer International Publishing, 2019. ISBN: 978-3-030-22101-0. DOI: 10.1007/978-3-030-22102-7 (cit. on pp. 2, 7, 21).

[Finger n.d.] Marcelo Finger. "Tractable Max-SAT in Graphic Description Logics". In preparation (cit. on pp. 8, 10, 11, 13, 15).

[Georgakopoulos *et al.* 1988] George Georgakopoulos, Dimitris Kavvadias, and Christos H Papadimitriou. "Probabilistic satisfiability". In: *Journal of complexity* 4.1 (1988), pp. 1–11 (cit. on p. 21).

[Gutiérrez-Basulto *et al.* 2011] Víctor Gutiérrez-Basulto, Jean Christoph Jung, Carsten Lutz, and Lutz Schröder. "A Closer Look at the Probabilistic Description Logic Prob-EL". In: *Proc. 25th Conference on Artificial Intelligence (AAAI-11)*. Ed. by Wolfram Burgard and Dan Roth. AAAI Press, 2011, pp. 197–202 (cit. on p. 21).

[Gutiérrez-Basulto *et al.* 2017] Víctor Gutiérrez-Basulto, Jean Christoph Jung, Carsten Lutz, and Lutz Schröder. "Probabilistic description logics for subjective uncertainty". In: *Journal of Artificial Intelligence Research* 58 (2017), pp. 1–66 (cit. on p. 21).

[Heinsohn 1994] Jochen Heinsohn. "Probabilistic description logics". In: *Uncertainty Proceedings 1994*. Elsevier, 1994, pp. 311–318 (cit. on p. 21).

[Hitzler *et al.* 2009] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F Patel-Schneider, Sebastian Rudolph, *et al.* "OWL 2 web ontology language primer". In: *W3C recommendation* 27.1 (2009), p. 123 (cit. on p. 5).

[Jaumard and Simeone 1987] Brigitte Jaumard and Bruno Simeone. "On the complexity of the maximum satisfiability problem for Horn formulas". In: *Information Processing Letters* 26.1 (1987), pp. 1–4 (cit. on p. 21).

[Krötzsch *et al.* 2012] Markus Krötzsch, Frantisek Simancik, and Ian Horrocks. "A description logic primer". In: *arXiv preprint arXiv:1201.4089* (2012) (cit. on p. 3).

[Lamy 2017] Jean-Baptiste Lamy. "Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies". In: *Artificial intelligence in medicine* 80 (2017), pp. 11–28 (cit. on p. 14).

[Lukasiewicz 2008] Thomas Lukasiewicz. "Expressive probabilistic description logics". In: *Artificial Intelligence* 172.6-7 (2008), pp. 852–883 (cit. on p. 21).

[Van Rossum and Drake 2009] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. isbn: 1441412697 (cit. on p. 13).