Probabilistic Abox Abduction in Description Logics

Murat Şensoy[1], Achille Fokoue[2], Mudhakar Srivatsa[2], and Jeff Z. Pan[1]

[1]Department of Computing Science, University of Aberdeen, UK
[2]IBM T. J. Watson Research Center, NY, USA

Abstract. *ABox* abduction is the process of finding statements that should be added to an ontology to entail a specific conclusion. In this paper, we propose an approach for probabilistic abductive reasoning for $\mathcal{SHIQ}$. Our evaluations show that the proposed approach significantly extends classical abduction by effectively and correctly estimating probabilities for abductive explanations. Lastly, based on the ideas proposed for $\mathcal{SHIQ}$, we describe a tractable algorithm for $\textsf{DL-Lite}_{\mathcal{R}}$.

1 Introduction

The prevailing framework for querying and reasoning over data on the semantic web has been based on logical deduction: the ability to infer and retrieve implied facts logically entailed from a knowledge base. A number of highly optimized deductive reasoners (*e.g.,* Pellet [20], KAON2 [8], Hermit [14], TrOWL [22]) have been developed in compliance with the Web Ontology Language standard (OWL) and successfully used in various applications (*e.g.,* Matching Patients to Clinical Trials [16]).

However, there is a growing realization in the semantic web community [5] that deduction is insufficient for new classes of applications that could leverage the increasing number of formal ontologies available on the semantic web and in knowledge rich domains such as heathcare and life sciences. Applications that seek to explain observations (*e.g.,* a patient's symptoms or errors or failures in a complex systems) or expectations that are not logically entailed from our current knowledge would greatly benefit from an abductive reasoning paradigm. Abductive reasoning is the process of finding the explanations for a set of observations. In this context, an explanation is a set of axioms $\mathcal{S}$ that, if added to a knowledge base $\mathcal{K}$, will ensure that the combined knowledge base ($\mathcal{K} \cup \mathcal{S}$) now logically entails the set of observations.

In many situations, the number of abductive explanations for an axiom could be very high, making it very costly to process all of these explanations. Various criteria have been considered in the literature to define a preference order on abductive explanations and to select the *best* explanations. Preferred explanations are typically the smallest in terms of either their number of axioms or according to subset inclusion. However, the smallest explanation in terms of size or set inclusion is not necessarily the most likely. Formal attempts to define the best logical abductive solutions in terms of their likelihood have traditionally required an explicit specification of a probabilistic model as part of the background knowledge [9, 17].

In this paper, we formalize the notion of likelihood of solutions w.r.t. to a background knowledge without extending the Description Logic (DL) formalism with a probabilistic model. We propose a novel approach for probabilistic *Abox* abduction for $\mathcal{SHIQ}$, one of the most expressive *DLs*. In this approach, instead of extending DL formalism with a probabilistic model, we rely on the ability to discover patterns of explanations in a background knowledge base and compute simple statistics to find the most prevalent patterns in the knowledge base. Then, we estimate the likelihoods of abductive explanations based on these statistics. Through empirical studies, we compare the proposed approach with non-probabilistic abduction where the abductive explanations are assumed equally likely. We show that the proposed approach can effectively estimate the likelihoods of the abductive explanations.

Reasoning in $\mathcal{SHIQ}$ is known to be intractable [1]. Our empirical studies also highlight the fact that abductive reasoning in $\mathcal{SHIQ}$ is computationally expensive. Based on the ideas proposed for $\mathcal{SHIQ}$, we have described a tractable algorithm for probabilistic abduction in $\mathsf{DL\text{-}Lite}_{\mathcal{R}}$ [2], the theoretical underpinning of OWL 2.0 QL profile. In particular, we show that the computational complexity class of this algorithm is the same as the computational complexity class of instance checking and conjunctive query answering in $\mathsf{DL\text{-}Lite}_{\mathcal{R}}$.

The remainder of the paper is organized as follows. Section 2 introduces preliminaries necessary to follow the paper. Section 3 describes a non-probabilistic abduction approach for $\mathcal{SHIQ}$ and Section 4 builds our probabilistic abduction for $\mathcal{SHIQ}$ upon it. Section 5 evaluates the proposed approach through empirical studies and Section 6 proposes a tractable approach for probabilistic abduction in $\mathsf{DL\text{-}Lite}_{\mathcal{R}}$. Lastly, Section 7 discusses the related work and Section 8 concludes the paper with an overview.

2 Preliminaries

2.1 $\mathcal{SHIQ}$ Description Logics

In this paper, unless stated otherwise, we consider ontologies of $\mathcal{SHIQ}$ expressiveness. In this section, we briefly introduce the semantics of $\mathcal{SHIQ}$, which is equivalent to OWL-DL 1.0 [1] minus nominals and datatype reasoning, as shown in Table 1 (We assume the reader is familiar with Description Logics [1]). Let $\mathcal{N}_C$ be the set of atomic concepts, $\mathcal{N}_R$ be the set of atomic roles, and $\mathcal{N}_I$ be the set of individuals. $\mathcal{N}_C$, $\mathcal{N}_R$, and $\mathcal{N}_I$ are mutually disjoint. Complex concepts and roles are built using constructs presented in Table 1(a).

A $\mathcal{SHIQ}$ knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consists of a Tbox $\mathcal{T}$ and an Abox $\mathcal{A}$. A Tbox $\mathcal{T}$ is a finite set of axioms, including:

– transitivity axioms of the form Trans(R) where R is a role.

[1] http://www.w3.org/2001/sw/WebOnt

Definitions	Semantics
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$\neg C$	$\Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$
$\exists R.C$	$\{x \vert \exists y. < x, y > \in R^{\mathcal{I}}, y \in C^{\mathcal{I}}\}$
$\forall R.C$	$\{x \vert \forall y. < x, y > \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$
$\leq nR$ C	$\{x \vert \, \vert\{< x, y > \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}\vert \leq n\}$
$\geq nR$ C	$\{x \vert \, \vert\{< x, y > \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}\vert \geq n\}$
R^{-}	$\{< x, y > \vert < y, x > \in R^{\mathcal{I}}\}$

(a) Constructors

Axioms	Satisfiability conditions
Trans(R)	$(R^{\mathcal{I}})^{+} = R^{\mathcal{I}}$
$R \sqsubseteq P$	$< x, y > \in R^{\mathcal{I}} \Rightarrow < x, y > \in P^{\mathcal{I}}$
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
$a : C$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
$R(a, b)$	$< a^{\mathcal{I}}, b^{\mathcal{I}} > \in R^{\mathcal{I}}$
$a \dot{\neq} b$	$a^{\mathcal{I}} \neq b^{\mathcal{I}}$

(b) Axioms

Table 1. SHIQ Description Logic

– role inclusion axioms of the form $R \sqsubseteq P$ where R and P are roles. $\sqsubseteq^{*}$ denotes the reflexive transitive closure of the $\sqsubseteq$ relation on roles.
– concept inclusion axioms of the form $C \sqsubseteq D$ where C and D are concept expressions.

An Abox $\mathcal{A}$ is a set of axioms of the form $a : C$, $R(a, b)$, and $a \dot{\neq} b$.

As for First Order Logic, a model theoretical semantic is adopted here. In the definition of the semantics of $\mathcal{SHIQ}$, $\mathcal{I}$= ($\Delta^{\mathcal{I}}$, $.^{\mathcal{I}}$) refers to an interpretation where $\Delta^{\mathcal{I}}$ is a non-empty set (the domain of the interpretation), and $.^{\mathcal{I}}$, the interpretation function, maps every atomic concept C to a set $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, every atomic role R to a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and every individual a to $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The interpretation function is extended to complex concepts and roles as indicated in the second column of Table 1(a).

An interpretation $\mathcal{I}$ is a model of a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, denoted $\mathcal{I} \models \mathcal{K}$, iff. it satisfies all the axioms in $\mathcal{A}$, and $\mathcal{T}$ (see Table 1(b)). A knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is consistent iff. there is a model of $\mathcal{K}$. Let α be an axiom, a knowledge base $\mathcal{K}$ entails α, denoted $\mathcal{K} \models \alpha$, iff. every model of $\mathcal{K}$ satisfies α.

2.2 Conjunctive Query

Given a knowledge base $\mathcal{K}$ and a set of variables $\mathcal{N}_V$ disjoint from $\mathcal{N}_I$, $\mathcal{N}_R$, and $\mathcal{N}_C$, a conjunctive query q is of the form $x_1, ..., x_n \leftarrow t_1 \wedge ... \wedge t_m$ where, for $1 \leq i \leq n$, $x_i \in \mathcal{N}_V$ and, for $1 \leq j \leq m$, t_j is a query term. A query term t is of the form $C(x)$ or $R(x, y)$ where x and y are either variables in $\mathcal{N}_V$ or individuals in $\mathcal{N}_I$, C is an atomic concept and R is an atomic role. $body(q)$ denotes the set of query terms of q. $Var(q)$ refers to the set of variables occurring in query q, and $DVar(q) = \{x_1, ..., x_n\}$ is the subset of $Var(q)$ consisting of distinguished (or answer) variables. Non-distinguished variables (i.e., variables in $Var(q) - DVar(q)$) are existentially quantified variables.

Let π be a total function from the set $DVar(q)$ of distinguished variables to the set $\mathcal{N}_I$ of individuals. We say that π is an answer to q in the interpretation

$\mathcal{I}$, denoted $\mathcal{I} \models q[\pi]$, if there exists a total function ϕ from $Var(q) \cup \mathcal{N}_I$ to $\Delta^{\mathcal{I}}$ such that the following hold:

- if $x \in DVar(q)$, $\phi(x) = \pi(x)^{\mathcal{I}}$
- if $a \in \mathcal{N}_I$, $\phi(a) = a^{\mathcal{I}}$
- $\phi(x) \in C^{\mathcal{I}}$ for all query terms $C(x) \in body(q)$.
- $(\phi(x), \phi(y)) \in R^{\mathcal{I}}$ for all query terms $R(x, y) \in body(q)$.

$ans(q, \mathcal{I})$ denotes the set of all answers to q in $\mathcal{I}$. π is said to be a certain answer to q over a knowledge base $\mathcal{K}$ iff. $\pi \in ans(q, \mathcal{I})$ for every model $\mathcal{I}$ of $\mathcal{K}$. The set of all certain answers of q over $\mathcal{K}$ is denoted $cert(q, \mathcal{K})$.

2.3 Abductive Reasoning

Abduction is an important reasoning service which provides possible explanations (or hypotheses) for observations that are not entailed by our current knowledge. In this section, we briefly formalize the notion of Abox abduction.

Definition 1 *An abox abduction problem is a tuple* $(\mathcal{K}, \mathcal{H}, A(a))$*, where* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *is a knowledge base, called the background knowledge base,* $\mathcal{H}$ *is a set of atomic concepts or roles,* A *is an atomic concept and* a *is an individual appearing in the Abox* $\mathcal{A}$ *of* $\mathcal{K}$ *such that* $\mathcal{K}$ *does not entail* $A(a)$*.*

In the previous definition, K corresponds to the background knowledge whose Tbox provides a conceptualization of the domain of discourse. $\mathcal{H}$ represents the concepts and roles that may appear in an explanation for the observation $A(a)$.

Definition 2 *A solution to an abox abduction problem* $\mathcal{P} = (\mathcal{K} = (\mathcal{T}, \mathcal{A}), \mathcal{H}, A(a))$ *is a set* $\mathcal{S} = \{C(u) | C \in \mathcal{H}, u \in \mathcal{N}_I\} \cup \{R(u, v) | R \in \mathcal{H}, (u, v) \in \mathcal{N}_I^2\}$ *of abox assertions such that:*

1. *The knowledge base* $(\mathcal{T}, \mathcal{A} \cup \mathcal{S})$ *is consistent*
2. $(\mathcal{T}, \mathcal{A} \cup \mathcal{S}) \models A(a)$

3 Non-Probabilistic Abduction for $\mathcal{SHIQ}$

Given an extensionally reduced $\mathcal{SHIQ}$ knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, the KAON2 transformation computes a disjunctive datalog program with equality, denoted by $DD(\mathcal{K})$. This datalog program is the union of a set of function-free rules compiled from $\mathcal{T}$ by exploiting certain resolution operations [8] and a set of ground rules directly translated from $\mathcal{A}$. Hustadt *et al.* showed that $\mathcal{K}$ is consistent if and only if $DD(\mathcal{K})$ is satisfiable [8]. The most state-of-the-art abduction systems [10, 13] are built on Prolog engines that work on plain datalog programs. Du *et al.* described a procedure to translate $DD(\mathcal{K})$ into a Prolog program $\bar{\mathcal{K}}$. That is, using a chain of transformation, we can convert $\mathcal{K}$ into a Prolog program $\bar{\mathcal{K}}$. Then, we can solve the abductive reasoning problem using $\bar{\mathcal{K}}$ and existing abductive reasoning methods for plain datalog programs [4]. In this section, we exploit this approach to find all abductive explanations for $A(a)$, then in Section 4 we propose an approach to estimate likelihoods for these explanations.

Figure 1 shows a simplified Prolog program for abductive reasoning over $\bar{\mathcal{K}}$. This simple program is composed of six rules. Using only six rules, this program defines the predicate $abduce(A, \mathcal{S})$, where A is an axiom such as $'Alcoholic'('Victoria')$ and $\mathcal{S}$ is a solution to the abduction problem (i.e., abductive explanation) computed by the program. For this purpose, it simply starts with an empty set of axioms as shown in the rule 1 and populates it iteratively based on other rules. The rule 2 guarantees that already entailed Abox axioms do not appear in $\mathcal{S}$. The rule 4 expands a complex abox axiom into its components by finding a clause in $\bar{\mathcal{K}}$ so that the head of the clause unifies the axiom. The rule 5 prevents redundancies in the solution. The rule 6 expands an existing partial solution by adding a new axiom if this axiom is an *abducible* and this addition does not create an inconsistency. Abducibles correspond to $\mathcal{H}$, i.e., the concepts or roles that we desire to appear in the solution. If a concept or role does not appear in the head of any clause in the prolog knowledge base, then it should also be an abducible. We may note that concept or role expressions in $\mathcal{K}$ may results in cycles in $\bar{\mathcal{K}}$. For instance, an expression such as $\exists hasParent.human \sqsubseteq human$ leads to a Prolog clause $human(X) :- hasParent(X, Y), human(Y)$. For the sake of simplicity, we have not shown it in Figure 1, but we implemented rule 4 so that it does not expand an axiom if this expansion results in a loop, instead this axiom is added directly to the solution. In this way, we prevent infinite loops during abductive reasoning.

```
1. abduce(A,S):-
      abduce(A,[],S).
2. abduce(A,S,S):-
      holds(A),!.
3. abduce((A,B),S0,S):-!,
      abduce(A,S0,S1),
      abduce(B,S1,S).
4. abduce(A,S0,S):-!,
      clause(A,B),
      abduce(B,S0,S).
5. abduce(A,S,S) :-
      member(A,S),!.
6. abduce(A,S,[A|S]):-
      abducible(A),
      checkConsistency([A|S]).
```

Fig. 1. Simplified abductive reasoner for plain datalog programs.

4 Probabilistic Abduction for $\mathcal{SHIQ}$

Various minimality criteria have been considered in the literature to define a preference order on solutions to an abduction problem. Preferred solutions are typically the smallest in terms of either their number of axioms or according to subset inclusion. However, the smallest explanation in terms of size or set inclusion is not necessarily the most likely. Formal attempts to define the best logical abductive solutions in terms of their likelihood have traditionally required an explicit specification of a probabilistic model as part of the background knowledge [9, 17]. In this section, we formalize the notion of likelihood of solutions w.r.t. to the background knowledge without extending the DL formalism with a probabilistic model. First, we introduce, as a running example, the following knowledge base $K = (\mathcal{T}, \mathcal{A})$:

Example 1. $\mathcal{T} = \{\exists addictedTo.AlcoholicBeverage \sqsubseteq Alcoholic,$
$Wine \sqsubseteq AlcoholicBevarage, Whisky \sqsubseteq AlcoholicBevarage,$
$Vodka \sqsubseteq AlcoholicBevarage, RedWine \sqsubseteq Wine, WhiteWine \sqsubseteq Wine$
$Man \sqsubseteq Person, Woman \sqsubseteq Person, Man \sqsubseteq \neg Woman\}$
$\mathcal{A} = \{addictedTo(Mary, red_1), addictedTo(Helen, red_2), addictedTo(Jane, white_1)$
$addictedTo(Elisabeth, vodka_1), addictedTo(Elisabeth, whisky_1),$
$addictedTo(John, vodka_2), addictedTo(Paul, vodka_1), addictedTo(Henry, vodka_2),$
$addictedTo(Bob, vodka_3), addictedTo(James, whisky_1),$
$WhiteWine(white_1), Whisky(whisky_1), Woman(Victoria)\}$
$\bigcup\{RedWine(red_n) | 1 \leq n \leq 2\} \bigcup \{Vodka(vodka_n) | 1 \leq n \leq 3\}$

The following are valid solutions to the abduction problem $\mathcal{P} = (K, \mathcal{H} = \{addictedTo, Vodka, Wine, RedWine, WhiteWine, Whisky\}, Alcoholic(Victoria))$:

$$\mathcal{S}_1 = \{addictedTo(Victoria, vodka_1)\}$$
$$\mathcal{S}_2 = \{addictedTo(Victoria, newWine), Wine(newWine)\}$$

Which of the two explanations is more likely given the background knowledge base? In $(\mathcal{T}, \mathcal{A} \cup \mathcal{S}_1)$, the only abox justification[2] for $Alcoholic(Victoria)$, i.e. a minimum set of Abox assertions $\mathcal{J}$ such that $(\mathcal{T}, \mathcal{J}) \models Alcoholic(Victoria)$, is $\mathcal{J}_1 = \{addictedTo(Victoria, vodka_1), Vodka(vodka_1)\}$, whereas $\mathcal{J}_2 = \{addictedTo(Victoria, newWine), Wine(newWine)\}$ is the only justification for $Alcoholic(Victoria)$ in $(\mathcal{T}, \mathcal{A} \cup \mathcal{S}_2)$. Now, the justification $\mathcal{J}_1$ can be abstracted into a pattern of justifications $\widehat{\mathcal{J}_1} = x \leftarrow addicted(x, y) \wedge Vodka(y)$ representing all justifications of $Alcoholic(x)$ involving an addiction to a $Vodka$. In the background knowledge base K, 5 out of 10 justifications for $Alcoholic(x)$, with x an individual in $\mathcal{A}$, are instances of the pattern $\widehat{\mathcal{J}_1}$. Intuitively, a justification $\mathcal{J}$ for $Alcoholic(a)$, where a is an individual in $\mathcal{A}$, is an instance of the pattern $\widehat{\mathcal{J}_1} = x \leftarrow addictedTo(x, y) \wedge Vodka(y)$ iff. the conjunctive query $x \leftarrow addictedTo(x, y) \wedge Vodka(y)$ issued over $(\mathcal{T}, \mathcal{J})$ has a as an answer. On the other hand, only 3 out of 10 justifications for $Alcoholic(x)$ in the background KB K are instances of the justification pattern $\widehat{\mathcal{J}_2} = x \leftarrow addictedTo(x, y) \wedge Wine(y)$ associated with $\mathcal{J}_2$. The likelihood of the solution $\mathcal{S}_1$ (resp. $\mathcal{S}_2$)), denoted $Pr(\mathcal{S}_1)$ (resp. $Pr(\mathcal{S}_2)$), is 0.5 (resp. 0.3). Thus, $\mathcal{S}_1$ appears as the most likely solution.

Next, we formally define the notions of a justification pattern and an instance of a justification pattern.

Definition 3 *Let $\mathcal{B}$ be a subset of the Abox $\mathcal{A}$ of a knowledge base $\mathcal{K}$ and a be an individual in $\mathcal{B}$. The abox pattern with focus a, denoted $\widehat{\mathcal{B}(a)}$, associated with $\mathcal{B}$*

[2] An abox justification for an axiom is a minimal set of a-box assertions entailing it.

is the conjunctive query $x \leftarrow t_1 \wedge ... \wedge t_n$ *such that* $\{t_1, ..., t_n\} = \{A(\pi(b))|A(b) \in \mathcal{B}\} \cup \{R(\pi(b), \pi(c))|R(b, c) \in \mathcal{B}\}$*, where* π *is an injective mapping from individuals in* $\mathcal{B}$ *to new variables such that* $\pi(a) = x$*.*

Definition 4 *Given a knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$*, a subset* $\mathcal{S}$ *of* $\mathcal{A}$ *containing the individual* b *is an instance with focus* b *of an abox pattern* $q = x \leftarrow t_1 \wedge ... \wedge t_n$*, denoted* $\mathcal{S} \models^b q$*, iff.* $(x \rightarrow b) \in cert(q, (\mathcal{T}, \mathcal{S}))$*.*

Notation 1 $\Omega(K, A(a))$ *denotes the set of all abox justifications for* $A(a)$ *in* $K = (\mathcal{T}, \mathcal{A})$*, where* A *is an atomic concept and* a *is an individual in* $\mathcal{A}$*. Then,* $|\Omega(\mathcal{K}, A|B)|$ *is the number of all justifications in the background knowledge base* $\mathcal{K}$ *for answers of the conjunctive query* $x \leftarrow A(x)$ *that are also instances of the concept* B*. Given* $Ind(\mathcal{A})$ *denotes the set of individuals in* $\mathcal{A}$*,* $|\Omega(\mathcal{K}, A|B)|$ *is computed as follows.*

$$|\Omega(\mathcal{K}, A|B)| = \sum_{a \in Ind(\mathcal{A}) \ s.t. \ \mathcal{K} \models B(a)} |\Omega(\mathcal{K}, A(a))|$$

Let $\mathcal{S}$ be a solution to an abduction problem $(\mathcal{K} = (\mathcal{T}, \mathcal{A}), \mathcal{H}, A(a))$, computed on $\bar{\mathcal{K}}$ as described in Section 3. Assuming that $\mathcal{J}$ is the only justification for $A(a)$ in $\mathcal{A} \cup \mathcal{S}$, the likelihood of the solution $\mathcal{S}$ is intuitively the fraction of justifications for the answers to the query $x \leftarrow A(x)$ in the background KB $\mathcal{K}$ that conform to (ie. are instances of) the abox pattern associated with $\mathcal{J} \cup \mathcal{S}$. Now, in Definition 5, we define how unconditional likelihoods for the solutions could be computed, where the number of justifications in the background KB, $|\Omega(\mathcal{K}, A)|\top|$, is a measure of our confidence in the computed likelihood. Here, the unconditional likelihood of an explanation $\mathcal{S}$ is formalised as the likelihood of the most probable justification of $A(a)$ in $\mathcal{K} \cup S$. Let $\mathcal{J}$ be a justification of $A(a)$, then the probability of $\mathcal{J}$, denoted as $Pr_{just}(\mathcal{J})$, is computed as frequency of the justification pattern derived from $\mathcal{J}$ in all abox justifications for the current instances of A, i.e., $\Omega(\mathcal{K}, A|\top)$. Consider Example 1, the likelihood of $\mathcal{S}_1$ is higher than that of $\mathcal{S}_2$, since addiction to vodka is more frequent than addiction to wine among all known alcoholics.

Definition 5 *The unconditional likelihood of a solution* $\mathcal{S}$ *of an abduction problem* $(\mathcal{K} = (\mathcal{T}, \mathcal{A}), \mathcal{H}, A(a))$*, denoted* $Pr(\mathcal{S})$*, is the real number between 0 and 1 defined as follows:*

- *if* $cert(x \leftarrow A(x), \mathcal{K}) = \emptyset$*,* $Pr(\mathcal{S}) = 0$
- *if* $cert(x \leftarrow A(x), \mathcal{K}) \neq \emptyset$*,*
  $Pr(\mathcal{S}) = \max_{\mathcal{J} \in \Omega((\mathcal{T}, \mathcal{A} \cup \mathcal{S}), A(a))} Pr_{just}(\mathcal{J})$
  $Pr_{just}(\mathcal{J}) = \frac{\sum_{b \in ind(\mathcal{A})} |\{\mathcal{J}' | \mathcal{J}' \in \Omega(\mathcal{K}, A(b)) \ and \ \mathcal{J}' \models^b \widehat{(\mathcal{J} \cup \mathcal{S})(a)}\}|}{|\Omega(\mathcal{K}, A|\top)|}$

Following Definition 5, let us note that if S has an axiom not entailed by any justification for $A(b)$, with b an individual in $\mathcal{A}$, $Pr(S) = 0$. This follows from the fact that no justification $\mathcal{J}'$ in $\mathcal{K}$ will be an instance of the pattern,

$\widehat{(\mathcal{J} \cup \mathcal{S})}(a)$, associated with $\mathcal{J} \cup \mathcal{S}$, where $\mathcal{J}$ is a justification in $(\mathcal{T}, \mathcal{A} \cup \mathcal{S})$ for $A(a)$. For example, $Pr(S_1 \cup \{Person(Victoria)\}) = 0$, where S_1 is the first solution introduced in the running example.

While computing likelihoods for solutions it is key to compute $\Omega(\mathcal{K}, A(a))$. Now we briefly describe how to compute it. Given an arbitrary primitive concept C and individual i in $\mathcal{K}$, $\mathcal{K} \models C(i)$ iff. $\bar{\mathcal{K}} \models C(i)$, so in order to efficiently compute abox justifications for $C(i)$, we use a Prolog meta-interpreter [21] that tracks the steps while proving $C(i)$ in $\bar{\mathcal{K}}$. Hence, using the meta interpreter, we enumerate all proofs for $A(i)$ in $\bar{\mathcal{K}}$, each of which is one justification. Based on these justifications, we compute $\Omega(\mathcal{K}, A(i))$.

Now, suppose the set of axioms $\{Woman(Mary), Woman(Helen), Woman(Jane)$, $Woman(Elisabeth)$, $Man(John)$, $Man(Paul)$, $Man(Henry)$, $Man(Bob)$, $Man(James)\}$ were added to the Abox $\mathcal{A}$ of Example 1. It could be argued that the most likely solution for $\mathcal{P}$, is $\mathcal{S}_2 = \{\ addictedTo(Victoria, newWine)$, $Wine(newWine)\}$ instead of $\mathcal{S}_1 = \{addictedTo(Victoria, vodka_1\}$ because, when we consider only female alcoholics, 3 out of 4 are addicted to Wine and Victoria is known to be a woman. Assuming that we have enough instances of the concept $Woman$ in the background KB, a more appropriate measure is the likelihood of a solution to $\mathcal{P}$ knowing that $Victoria$ is a $Woman$. The following definition formalizes the notion of conditional likelihood.

Definition 6 *The conditional likelihood of a solution $\mathcal{S}$ of an abduction problem $(\mathcal{K} = (\mathcal{T}, \mathcal{A}), \mathcal{H}, A(a))$ knowing a is an instance of a concept C, denoted $Pr(\mathcal{S}|C)$, is the real number between 0 and 1 defined as follows:*

- *if $cert(x \leftarrow A(x) \wedge C(x), \mathcal{K}) = \emptyset$, $Pr(\mathcal{S}) = 0$*
- *if $cert(x \leftarrow A(x) \wedge C(x), \mathcal{K}) \neq \emptyset$,*
  $Pr(\mathcal{S}|C) = \max_{\mathcal{J} \in \Omega((\mathcal{T}, \mathcal{A} \cup \mathcal{S}), A(a))} Pr_{just}(\mathcal{J}|C)$
  $Pr_{just}(\mathcal{J}|C) = \frac{\sum_{b \in ind(\mathcal{A}) \text{ and } \mathcal{K} \models C(b)} |\{\mathcal{J}' | \mathcal{J}' \in \Omega(\mathcal{K}, A(b)) \text{ and } \mathcal{J}' \models^b \widehat{(\mathcal{J} \cup \mathcal{S})}(a)\}|}{|\Omega(\mathcal{K}, A|C)|}$

Definition 6 allows us to estimate the likelihood of abductive explanations (i.e., solutions for the abduction problem) for $A(a)$ within the context that we know a is an instance of the concept C. If C is too specific, then it significantly reduce our confidence on $Pr(\mathcal{S}|C)$ by reducing $|\Omega(\mathcal{K}, A|C)|$. For instance, if C is a concept with only three instances $\{a, b, c\}$ in $\mathcal{K}$, then $|\Omega(\mathcal{K}, A|C)|$ can be much smaller than desired. To select the best context, here we propose starting from the most specific context and iteratively generalize it until our criteria for an acceptable context holds. This idea is formalized in the algorithm below. The algorithm accepts a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and the threshold N as input to find the most specific concept description C acceptable for serving as a context. Here, N determines the minimum number of individuals C should have to stop further generalization. In the algorithm, we keep a set S representing the set of C's direct super concepts. Initially, C is set to the most specific context C', which is the intersection of a's direct types (line 1), and S contains only C' (line 2). While C has a number of individuals less than N and S contains some elements, we get the most specific element $s \in S$ (line 4). The most specific element is a concept or a concept description having the longest path to the top concept $\top$ when put into the concept hierarchy derived from $\mathcal{K}$. Then, we update S by removing s and adding super concepts of s (line 5). Some super concepts of s may be equivalent to s, so we remove these while updating S. Lastly, at the end of each iteration, we set C to a DL concept description, the intersection of elements in S (line 6). If S contains only one element, C is set to this element. By selecting the most specific element at each iteration during generalization, we aim fine grained generalization. However, if the total number of individual in the ontology is less than N, S becomes empty after the removal of the top concept $\top$ and the generalization stops; the algorithm returns $\top$ in this situation.

findContext($\mathcal{K} = (\mathcal{T}, \mathcal{A}), a, N)$)
Input: $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ a knowledge base, a an individual, N a threshold
Output: C a concept description representing context
(1) $C = C' = intersectionOf(getDirectTypes(a))$
(2) $S = \{C'\}$
(3) **while** $|getIndividuals(C, \mathcal{K})| < N$ **and** $|S| > 0$
(4) $s = getMostSpecific(S, \mathcal{T})$
(5) $S = S \setminus \{s\} \cup supers(s, \mathcal{T}) \setminus equivalents(s, \mathcal{T})$
(6) $C = intersectionOf(S)$
(7) **return** C

The algorithm allows us to generalize context gradually until reaching a concept description C that has desirable number of instances. However, at each iteration, the distance between the new C and the most specific context C' may increase further.This brings the risk of having a context with enough number of instances, but failing represent a as expected. Once we define a distance metric between two concept descriptions C' and C, we can introduce another threshold δ for distance and avoid over generalization by testing $distance(C, C') < \delta$ during iterations, at line 3 of the algorithm. We can use various distance metrics, two of which can be summarized as: **i)** the number of iterations done to derive C from C' and **ii)** the distance between the concept descriptions C' and C in the concept hierarchy derived from $\mathcal{K}$, after inserting them as new concepts into $\mathcal{K}$ if they do not exist there already.

In Definition 6, the likelihood of a solution is defined in terms of justifications in the background knowledge base. Unfortunately, computing all justifications is well known to be intractable [11]. In Section 6, we show that abductive explanations and their likelihoods can be computed efficiently (PTime in the size of the TBox and LogSpace in the size of the Abox) for DL-Lite$_{\mathcal{R}}$ KBs.

5 Evaluation of Abduction in $\mathcal{SHIQ}$

In order to evaluate our approach, we have randomly created five $\mathcal{SHIQ}$ ontologies.Properties of these ontologies are listed in Table 2. Each of these ontologies contains 10 *target concepts*. We measure the performance of the proposed approach through these target concepts. Each target concept C has at least n possible patterns of justification.

For each individual I in ontology O, we select a target concept C. Then, among all justification patterns of C, we randomly select one pattern $\widehat{\mathcal{J}}$. Based on the selected justification pattern, we add new ABox axioms to O so that I would be an instance of C. For instance, if $\widehat{\mathcal{J}}$ is a pattern of three atoms such as $C(X) \leftarrow A(X), B(X,Y), D(Y)$, three ABox axioms $A(I)$, $B(I,i)$, and $D(i)$ are added to O, where i is another individual from O. We also extend the ABox by randomly adding other axioms about I, as long as these axioms do not lead to a second justification for $C(I)$. Let us note that justifications for instances of C in O are not uniformly distributed, because while selecting justification patterns of C for individuals, we use power low distribution instead of uniform distribution. This means that most of the justifications for C are instances of small number of justification patterns of C, while most justification patterns of C have a few or no instances in O.

We evaluate an abductive reasoning approach based on a target concept C as follows. First, we pick I, an instance of C. Let $\mathcal{J}$ be the justification for $C(I)$. Second, we randomly select a subset of the axioms in $\mathcal{J}$, denoted as Γ. Third, we remove the axioms in Γ from O. Hence, $C(I)$ does not hold any more. Fourth, using the abductive reasoning approach, we compute all abductive explanations of $C(I)$ for O with their probabilities. Let E_Γ be the abductive explanation unifying with Γ. Performance of the abduction approach is $Pr(E_\Gamma)$, which is the estimated probability of E_Γ by the abduction approach. Here, we compared three abductive reasoning approaches: non-probabilistic abduction (NPA), unconditional probabilistic abduction (UPA), and conditional probabilistic abduction (CPA). In NPA, after computing all abductive explanations, each explanation is considered equally likely, so if there are n explanations in total, each of them will have probability $1/n$. In CPA, we have used threshold $N = 20$ while generalizing context during abduction.

Table 2. Synthetic ontologies with different numbers of atomic concepts ($\#C$), roles ($\#R$), individual ($\#I$), $TBox$ axioms ($\#TA$), and $ABox$ axioms ($\#AA$).$\langle n \rangle$ denotes average number of justification patterns for main concepts.

Ontology	#C	#R	#I	#TA	#AA	$\langle n \rangle$
O_1	187	20	1000	339	4000	18
O_2	285	40	3732	602	12000	33
O_3	393	50	5199	719	13646	54
O_4	381	60	4795	722	17985	63
O_5	353	70	5283	691	23418	234

We have conducted experiments for 100 individuals in each ontology. Average values for our experiments are listed in Table 3, where $\langle \#E \rangle$ is the average number of explanations for $C(I)$; $\langle P_{NPA} \rangle$, $\langle P_{UPA} \rangle$, and $\langle P_{CPA} \rangle$ are the average performances of NPA, UPA, and CPA respectively; $\langle T_{NPA} \rangle$, $\langle T_{UPA} \rangle$, and $\langle T_{CPA} \rangle$ are average time spent in milliseconds by NPA, UPA, and CPA respectively. We can summarize our findings as follows. As the number of explanations increase, the performance of classical abduction decreases as expected. while the performance of UPA always significantly outperform NPA, it could not exceed 0.23 in the experiments. However, the performance of CPA is always around 0.8. These are the results for synthetic ontologies, where we have created abox axioms around a number of justification patterns. To test our approach using an ontology which is not created with justification patterns in mind, we have also conducted experiments using $Wine^-$ ontology, which is the W3C's Wine ontology [19] without nominals. Our results for $Wine^-$ endorse our findings based on the synthetic ontologies. That is, UPA and CPA significantly outperform NPA, i.e. at the magnitudes of 7 and 11 respectively. In general, during the computation of probabilities, CPA requires significantly more time than UPA does. Our analysis of time consumption highlights that the probabilistic abduction in $\mathcal{SHIQ}$ is expensive computationally as expected. In the following section, we propose a tractable algorithm for probabilistic abductive reasoning in $\mathsf{DL\text{-}Lite}_{\mathcal{R}}$.

Table 3. Results for synthetic ontologies and $Wine^-$ ontology ($N = 20$ for CPA).

Ontology	$\langle \#E \rangle$	$\langle P_{NPA} \rangle$	$\langle P_{UPA} \rangle$	$\langle P_{CPA} \rangle$	$\langle T_{NPA} \rangle$	$\langle T_{UPA} \rangle$	$\langle T_{CPA} \rangle$
O_1	15	0.073	0.23	0.84	73 ms.	850 ms.	1293 ms.
O_2	31	0.033	0.099	0.82	204 ms.	5612 ms.	7266 ms.
O_3	45	0.023	0.177	0.80	449 ms.	9643 ms.	12274 ms.
O_4	52	0.020	0.055	0.83	678 ms.	34183 ms.	35264 ms.
O_5	135	0.011	0.056	0.82	2658 ms.	70356 ms.	73938 ms.
$Wine^-$	70	0.014	0.099	0.16	37251 ms.	37288 ms.	46916 ms.

6 $\mathsf{DL\text{-}Lite}_{\mathcal{R}}$ Probabilistic Abduction Algorithm

In this section, we present a tractable algorithm to compute solutions to an abduction problem $\mathcal{P} = (\mathcal{K}, \mathcal{H}, A(a))$ along with their likelihood. The expressivity of background knowledge base $\mathcal{K}$ is restricted to $\mathsf{DL\text{-}Lite}_{\mathcal{R}}$ [2], the theoretical underpinning of OWL 2.0 QL profile. In particular, we show that the computational complexity class of this algorithm is the same as the computational complexity class of instance checking and conjunctive query answering in $\mathsf{DL\text{-}Lite}_{\mathcal{R}}$.

First, we remind the restrictions imposed by $\mathsf{DL\text{-}Lite}_{\mathcal{R}}$. Concepts and roles are formed according to the following syntax (A denotes an atomic concept and P an atomic role):

$$B \rightarrow A \mid \exists R \qquad\qquad R \rightarrow P \mid P^- \tag{1}$$

$$C \rightarrow B \mid \neg B \qquad\qquad E \rightarrow R \mid \neg R \tag{2}$$

Furthermore, Tbox axioms are restricted to the following forms: $B \sqsubseteq C$ and $R \sqsubseteq E$.

A key property of $\mathsf{DL\text{-}Lite}_{\mathcal{R}}$ is that conjunctive query answering in a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ can be reduced to union of conjunctive query answering against the KB $\mathcal{K}' = (\emptyset, \mathcal{A})$ with an empty Tbox. In other words, through query rewrite, the relevant part of $\mathcal{T}$ can be compiled into a new query. In [2], for a conjunctive query q and a Tbox $\mathcal{T}$, the result of the rewritting, denoted $PerfectRef(q, \mathcal{T})$, is a set of conjunctive queries such that, for any Abox $\mathcal{A}$, $cert(q, (\mathcal{T}, \mathcal{A})) = \bigcup_{q' \in PerfectRef(q,\mathcal{T})} cert(q', (\emptyset, \mathcal{A}))$.

For example, after removing the axiom $\exists addictedTo.AlcoholicBeverage \sqsubseteq Alcoholic$ from the Tbox of our running example, it becomes a DL-Lite$_{\mathcal{R}}$ KB. For the query $q = x \leftarrow AlcoholicBeverage(x)$,

$$PerfectRef(q, \mathcal{T}) = \{x \leftarrow AlcoholicBeverage(x), x \leftarrow Wine(x), x \leftarrow RedWine(x),$$

$$x \leftarrow WhiteWine(x), x \leftarrow Vodka(x), x \leftarrow Whisky(x)\}$$

Our approach to compute solutions to an abduction problem and their likelihood relies on the observation that an abox justification for $A(a)$ in a DL-Lite$_{\mathcal{R}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ must be an instance of the pattern formed by the body of one query in $PerfectRef(x \leftarrow A(x), \mathcal{T})$.

Proposition 1 *Let $K = (\mathcal{T}, \mathcal{A})$ be a DL-Lite$_{\mathcal{R}}$ knowledge base and A be an atomic concept such that $(\mathcal{T}, \emptyset)$ does not entail $\top \sqsubseteq A$. $\mathcal{J}$ is an abox justification for $A(a)$ in K iff. there exist $q \in PerfectRef(x \leftarrow A(x), \mathcal{T})$ and a mapping π from $Var(q) \cup Ind(\mathcal{J})$ to the set $Ind(\mathcal{J})$ of individuals in $\mathcal{J}$ such that (1) $\pi(x) = a$, (2) for $b \in Ind(\mathcal{J}), \pi(b) = b$, and*

$$(3)\ \mathcal{J} = \{C(\pi(u)) | C(u) \in body(q)\} \cup \{R(\pi(u), \pi(v)) | R(u, v) \in body(q)\}$$

Notation 2 *In the remainder of the paper, for a query q and a mapping π from $Var(q) \cup \mathcal{N}_I$ to $\mathcal{N}_I$, $construct(q, \pi)$ is the abox defined as follows:*

$$construct(q, \pi) = \{C(\pi(u)) | C(u) \in body(q)\} \cup \{R(\pi(u), \pi(v)) | R(u, v) \in body(q)\}$$

Proof.
- Suppose $\mathcal{J}$ is a abox justification for $A(a)$ in K. Since a is an answer to $x \leftarrow A(x)$, there exist $q \in PerfectRef(x \leftarrow A(x), \mathcal{T})$ and a mapping π from variables in q to individuals in $\mathcal{J}$ such that $\pi(x) = a$ and $construct(q, \pi) \subseteq \mathcal{J}$. π is extended to individuals b in $\mathcal{J}$ ($\pi(b) = b$). Since $construct(q, \pi)$ entails $A(a)$ and $\mathcal{J}$ is an abox justification for $A(a)$, it follows that $construct(q, \pi) = \mathcal{J}$.
- Let us assume that there exist $q \in PerfectRef(x \leftarrow A(x), \mathcal{T})$ and a mapping π from $Var(q) \cup Ind(\mathcal{J})$ to the set $Ind(\mathcal{J})$ of individuals in $\mathcal{J}$ satisfying the three conditions of Proposition 1. The rewritting performed by $PerfectRef(p, \mathcal{T})$ is such that every generated query $p' \in PerfectRef(q, \mathcal{T})$ has at most the same number of atoms as p and has at least one atom. Therefore $|construct(q, \pi)| = 1$, which makes $construct(q, \pi)$ mininal since $(\mathcal{T}, \emptyset)$ does not entail $\top \sqsubseteq A$. Thus, $construct(q, \pi)$ is an abox justification for $A(a)$.

Algorithm COMPUTESOLUTIONS computes a set of canonical solutions. Those solutions are canonical in the sense that, as shown in Theorem 1, an explanation for a solution not returned by COMPUTESOLUTIONS is always an instance of the pattern formed by a solution returned by COMPUTESOLUTIONS. Algorithm computeSolutions invokes Algorithm computeOmega to compute in $|\Omega(\mathcal{K}, A|C)|$. Before formally presenting properties of Algorithm computeSolutions, we briefly introduce below an important notation used in computeSolutions.

Notation 3 *For a conjunctive query q, the query $\overline{q}$ is the conjunctive query with the same body as q, but whose set of distinguished variables consists of all variables in q (i.e., $DVar(\overline{q}) = Var(q)$): $\overline{q} = x_1, ..., x_k \leftarrow t_1 \wedge ... \wedge t_m$ where $t_j \in body(q)$ for $1 \leq j \leq m$, and $x_1, ..., x_k$ are all variables in $body(q)$. Example, if $q = x \leftarrow A(x) \wedge R(x, y) \wedge S(y, z)$, then $\overline{q} = x, y, z \leftarrow A(x) \wedge R(x, y) \wedge S(y, z)$.*

computeSolutions($\mathcal{P} = (\mathcal{K} = (\mathcal{T}, \mathcal{A}), \mathcal{H}, A(a)), C$)
Input: $\mathcal{P} = (\mathcal{K}, \mathcal{H}, A(a))$ a abduction problem, C is a concept description or $\top$ s.t. $\mathcal{K} \models C(a)$
Output: set of pairs $(\mathcal{S}, p)$, where $\mathcal{S}$ is a solution to $\mathcal{P}$ with an conditional likelihood p knowing a is an instance of C
(1) $\Omega \leftarrow$ computeOmega($\mathcal{K}, A, C$)
(2) **foreach** q_i **in** $PerfectRef(x \leftarrow A(x), \mathcal{T})$
(3) π a mapping from $Var(q_i) \cup \mathcal{N}_I$ to $\mathcal{N}_I$ s.t. (1) $\pi(x) = a$, (2) for $b \in \mathcal{N}_I, \pi(b) = b$, and, (3) for $y \in Var(q_i)$ such that $y \neq x$, $\pi(y)$ is a new individual not present in $\mathcal{K}$
(4) $\mathcal{S} \leftarrow \{C(\pi(u)) | C(u) \in q_i\} \cup \{R(\pi(u), \pi(v)) | R(u, v) \in q_i\}$
(5) **if** $(\mathcal{T}, \mathcal{A} \cup \mathcal{S})$ is consistent and concepts and roles in S are all in $\mathcal{H}$
(6) $\omega \leftarrow |\{\pi' | \pi' \in cert(\overline{q_i}, (\emptyset, \mathcal{A}))\} \wedge \mathcal{K} \models C(\pi'(x))|$
(7) **emitSolution(** $(\mathcal{S}, \omega/\Omega)$ **)**

computeOmega($\mathcal{K} = (\mathcal{T}, \mathcal{A}), A, C$)
Input: $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ a DL-Lite$_{\mathcal{R}}$ knowledge base, A is an atomic concept, C is a concept description or $\top$
Output: $|\Omega(\mathcal{K}, A|C)|$
(1) $pref \leftarrow PerfectRef(x \leftarrow A(x), \mathcal{T})$
(2) $r \leftarrow 0$
(3) **foreach** q_i **in** $pref$
(4) **foreach** π **in** $cert(\overline{q_i}, (\emptyset, \mathcal{A}))$
(5) $new \leftarrow true$
(6) $\mathcal{J} \leftarrow construct(q_i, \pi)$
(7) // (8)-(10) ensure that $\mathcal{J}$ is not counted if it was previously discovered by q_j for $j < i$
(8) **foreach** q_j **in** $pref$ **s.t.** $j < i$
(9) **if** $\{\pi' | \pi' \in cert(\overline{q_j}, (\emptyset, \mathcal{J}))$ and $\pi'(x) = \pi(x)\} \neq \emptyset$
(10) $new \leftarrow false$
(11) **if** new and $\mathcal{K} \models C(\pi(x))$
(12) $r \leftarrow r + 1$
(13) **return** r

Theorem 1 *Let $\mathcal{P} = (\mathcal{K}, \mathcal{H}, A(a))$ be a abductive problem such that $K = (\mathcal{T}, \mathcal{A})$ a DL-Lite KB. Let C be a concept description or the top concept ($\top$). Algorithm* *computeSolutions*$(\mathcal{P}, C)$ *terminates. Furthermore, if* $\mathcal{S}_0$ *is a solution to* $\mathcal{P}$*, then, for each* $\mathcal{J} \in \Omega((\mathcal{T}, \mathcal{A} \cup \mathcal{S}_0), A(a))$*, there is* $(\mathcal{S}, p) \in$ *computeSolutions*$(\mathcal{P}, C)$ *such that the following hold:*

1. $Pr_{just}(\mathcal{J}|C) \leq p$ *(i.e. the conditional likelihood of* $\mathcal{J}$ *in* K *is less than or equal to p)*
2. $\mathcal{J} \models^a \widehat{\mathcal{S}(a)}$ *(i.e.* $\mathcal{J}$ *is an instance of the pattern formed by* $\mathcal{S}$*)*

The proof of Theorem 1 is a consequence of the Proposition 1.

Theorem 2 *Let* $\mathcal{P} = (\mathcal{K}, \mathcal{H}, A(a))$ *be a abductive problem such that* $K = (\mathcal{T}, \mathcal{A})$ *a DL-Lite KB. Let* C *be a concept description or the top concept* ($\top$)*. Algorithm* *computeSolutions*$(\mathcal{P}, C)$ *is* PTime *in the size of the TBox, and* LogSpace *in the size of the ABox (data complexity).*

Proof. The proof follows from the following properties of DL-Lite$_{\mathcal{R}}$ established in [2]:

– Consistency check and instance checking (*i.e.,* checking $\mathcal{K} \models C(b)$ for an individual b) in DL-Lite$_{\mathcal{R}}$ is PTime in the size of the TBox, and LogSpace in the size of the ABox.
– Conjunctive query answering against a KB with an empty Tbox is LogSpace in the size of the ABox (i.e., same complexity as conjunctive query answering against a database)
– For a conjunctive query q and a Tbox $\mathcal{T}$, the maximum size of $PerfectRef(q, \mathcal{T})$ is $(m(n+1)^2)^n$, where m is the size of the Tbox and n the size of the query q (i.e., the number of atoms in $body(q)$). Therefore $|PerfectRef(x \leftarrow A(x), \mathcal{T})| \leq 4 \times m$
– For a conjunctive query q and a Tbox $\mathcal{T}$, if $q' \in PerfectRef(q, \mathcal{T})$ then the number of atoms in q' is at most the same as the number of atoms in q. Therefore, if $q' \in PerfectRef(x \leftarrow A(x), \mathcal{T})$ then q' has at most one atom.

7 Related Work

Abduction in logic programming without probabilities has attracted a lot of attention, and several algorithms, including meta-interpreters written in Prolog, have been made [10, 13]. However, probabilistic abductive logic programming has not been studied nearly to the same extent. Poole proposed a probabilistic abduction approach for horn logic [17]. This approach considers a logic programming approach that uses a mix between depth-first and branch and bound search strategies for abduction where the probabilities are considered and only the most likely explanations are generated. Henning has proposed an approach for probabilistic abductive Logic programming with constraint handling rules [3]. This approach differs from other approaches to probabilistic logic programming by having both interaction with external constraint solvers and integrity constraints. Henning used probabilities to optimize the search for explanations using Dijkstra's shortest path algorithm. Hence, the approach explores always the most probable direction, so that investigation of less probable alternatives is suppressed or postponed. For plan recognition tasks represented in datalog, Raghavan and Mooney proposed to use Bayesian networks while estimating probabilities of abductive explanations [18]. They suggest to learn a Bayesian network from abductive explanations using structure learning techniques. Once the network structure is determined, the parameters of the network are learned using an external training set.

There are only a few works in the literature for abductive reasoning in DLs. However, unlike our approach, none of these works estimates probabilities for the computed abductive explanations. For $TBox$ abduction, Hubauer *et al.* proposed automata-based approach [7] while Noia *et al.* proposed an approach exploiting tableaux algorithms for DLs [15]. For $ABox$ abduction, Perald *et al.* proposed an approach based on a backward inference method [6]. It restricts axioms in the DL-based ontology to some special forms and does not use a notion of minimality for abductive solutions. Klarman *et al.* have proposed an approach for $ABox$ abduction in $\mathcal{ALC}$ fragment of OWL DL [12], but this approach cannot guarantee termination. Du *et al.* have propose another approach which is based on translation of $\mathcal{SHIQ}$ into Prolog and making abductive reasoning using existing a approaches for plain datalog programs [4]. They have showed that their guarantees termination and certain minimality of results. We have followed the same approach to enumerate all abductive explanations for an $ABox$ axiom.

8 Conclusions

In this paper, first we formalize probabilistic ABox abduction problem in DL. Then, we have proposed an approach for estimating likelihoods of abductive explanations. The proposed approach exploits the frequencies of justification patterns in $ABox$ within the context of specific individuals in a knowledge base. Our evaluations show that the proposed approach significantly outperform classical abduction approach where each explanation is assumed equally likely. Our findings also highlight that the probabilistic abduction in $\mathcal{SHIQ}$ is costly, as expected. That is why, we have presented a tractable algorithm for $\mathsf{DL\text{-}Lite}_{\mathcal{R}}$ at the end. As a future work, we plan to study the strength and weaknesses of the proposed approach extensively using various benchmarks.

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook*. Cambridge University Press, 2003.
2. D. Calvanese, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The dl-lite family. *J. of Automated Reasoning*, pages 385–429, 2007.
3. H. Christiansen. Constraint handling rules. chapter Implementing Probabilistic Abductive Logic Programming with Constraint Handling Rules, pages 85–118. 2008.

4. J. Du, G. Qi, Y.-D. Shen, and J. Z. Pan. Towards practical abox abduction in large owl dl ontologies. In *The Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI-11)*, San Francisco, USA, August 2011.
5. C. Elsenbroich, O. Kutz, and U. Sattler. A case for abductive reasoning over ontologies. In *OWLED'06*, pages –1–1, 2006.
6. S. Espinosa Perald, A. Kaya, S. Melzer, R. Moller, and M. Wessel. Towards a media interpretation framework for the semantic web. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, WI '07, pages 374–380, 2007.
7. T. Hubauer, S. Lamparter, and M. Pirker. Automata-based abduction for tractable diagnosis. In *Description Logics*, 2010.
8. U. Hustadt, B. Motik, and U. Sattler. Reasoning in description logics by a reduction to disjunctive datalog. *J. Autom. Reason.*, 39:351–384, October 2007.
9. P. D. R. K. James Blythe, Jerry R. Hobbs and R. J. Mooney. Implementing weighted abduction in markov logic. In *International Conference on Computational Semantics*, 2011.
10. A. C. Kakas, B. Van Nuffelen, and M. Denecker. A-system: problem solving through abduction. In *Proceedings of the 17th international joint conference on Artificial intelligence (IJCAI'01)*, pages 591–596, 2001.
11. A. Kalyanpur. *Debugging and Repair of OWL-DL Ontologies*. PhD thesis, University of Maryland, 2006.
12. S. Klarman, U. Endriss, and S. Schlobach. Abox abduction in the description logic $\mathcal{ALC}$. *J. Autom. Reason.*, 46:43–80, 2011.
13. P. Mancarella, G. Terreni, F. Sadri, F. Toni, and U. Endriss. The CIFF proof procedure for abductive logic programming with constraints: Theory, implementation and experiments. *Theory Pract. Log. Program.*, 9:691–750.
14. B. Motik, R. Shearer, and I. Horrocks. Hypertableau Reasoning for Description Logics. *Journal of Artificial Intelligence Research*, 36:165–228, 2009.
15. T. D. Noia, E. D. Sciascio, and F. M. Donini. A tableaux-based calculus for abduction in expressive description logics: Preliminary results. In *Description Logics*, 2009.
16. C. Patel, J. J. Cimino, J. Dolby, A. Fokoue, A. Kalyanpur, A. Kershenbaum, L. Ma, E. Schonberg, and K. Srinivas. Matching patient records to clinical trials using ontologies. In *ISWC/ASWC*, pages 816–829, 2007.
17. D. Poole. Probabilistic horn abduction and bayesian networks. In *Artificial Intelligence*, 1993.
18. S. Raghavan and R. Mooney. Bayesian abductive logic programs. In *Proceedings of the AAAI Workshop on Statistical Relational AI*, pages 82–87, 2010.
19. E. Sirin, B. Cuenca Grau, and B. Parsia. From wine to water: Optimizing description logic reasoning for nominals. In *Proceedings of KR-2006*, pages 90–99, 2006.
20. E. Sirin and B. Parsia. Pellet: An owl dl reasoner. In *Description Logics*, 2004.
21. L. Sterling and L. U. Yalcinalp. Explaining prolog based expert systems using a layered meta-interpreter. In *Proceedings of the 11th international joint conference on Artificial intelligence*, pages 66–71, 1989.
22. E. Thomas, J. Z. Pan, and Y. Ren. TrOWL: Tractable OWL 2 Reasoning Infrastructure. In *the Proc. of the Extended Semantic Web Conference (ESWC2010)*, 2010.