A Generic Framework for Description Logics with Uncertainty

Volker Haarslev, Hsueh-Ieng Pai, and Nematollaah Shiri

Concordia University
Dept. of Computer Science & Software Engineering
Montreal, Quebec, Canada
{haarslev, hsueh_pa, shiri}@cse.concordia.ca

Abstract. We propose an extension to Description Logics (DLs) with uncertainty which unifies and/or generalizes a number of existing frameworks for DLs with uncertainty. To this end, we first give a classification of these frameworks and identify the essential features as well as properties of the various combination functions allowed in the underlying uncertainty formalisms they model. This also allows us express the semantics of the DL elements in a flexible manner. We illustrate how various DLs with uncertainty can be expressed in our generic framework.

1 Introduction

Ever since Tim Berners-Lee introduced the vision of the Semantic Web [2], attempts have been made on making Web resources more machine-interpretable by giving Web resources a well-defined meaning through semantic markups. One way to encode such semantic markups is using ontologies. Over the last few years, a number of ontology languages have been developed, most of which have a foundation based on *Description Logics* (DLs) [1]. The family of DLs is a subset of first-order logic (FOL) that is considered to be attractive as it keeps a good compromise between expressive power and computational tractability.

Uncertainty management has been a challenge for over two decades in database (DB) and artificial intelligence (AI) research (see [10, 12]), and has recently attracted the attention of the DL community. *Uncertainty* is a form of deficiency or imperfection commonly found in the real-world information/data. A piece of information is uncertain if its truth is not established definitely.

Despite of the popularity of DLs, it has been realized that the standard DL framework is inadequate to model uncertainty. For example, in the medical domain, one might want to express that: "It is very likely that an obese person would have heart disease", where "obese" is a vague concept that may vary across regions or countries, and "likely" shows the uncertain nature of the knowledge. Such knowledge cannot be expressed within the current scope of DL.

Recently, a number of proposals have been put forward which extend DLs with uncertainty. Some of them deal with the vagueness aspect while others deal with the uncertainty aspect. We do not intend to compare which extension

is better. In fact, different applications may require using different aspects. It may even be desired in some cases to model different aspects within the same application.

Following the parametric approach proposed in [11], we propose a generic DL framework with uncertainty in this paper as a unifying umbrella for several existing frameworks for DLs with uncertainty. This not only provides a uniform access over knowledge that has been encoded using DL with various kinds of uncertainty, but also allows one to study various problems, including semantics, query processing and optimization, design, and implementation of such frameworks in a framework-independent manner.

The rest of this paper is organized as follows. Section 2 presents a classification of existing frameworks of uncertainty in DL. In section 3, we present our generic framework for DL with uncertainty in detail, along with examples illustrating how it can represent uncertainty modeled in the frameworks considered. Concluding remarks and some future directions are presented in section 4.

2 Approaches to DL with Uncertainty

On the basis of their mathematical foundation and the type of uncertainty modeled, we can classify existing proposals of DLs with uncertainty into three approaches: fuzzy, probabilistic, and possibilistic approach.

The fuzzy approach, based on fuzzy set theory [19], deals with the vagueness in the knowledge, where a proposition is true to only some degree. For example, the statement "Jason is obese with degree 0.4" indicates Jason is slightly obese. Here, the value 0.4 is the degree of membership that Jason is in concept obese.

The probabilistic approach, based on the classical probability theory, deals with the uncertainty due to lack of knowledge, where a proposition is either true or false, but one does not know for sure which one is the case. Hence, the certainty value refers to the probability that the proposition is true. For example, one could state that: "The probability that Jason would have heart disease given that he is obese lies in the range $[0.8, 1]$."

Finally, the possibilistic approach, based on possibility theory [20], allows both certainty (necessity measure) and possibility (possibility measure) be handled in the same formalism. For example, by knowing that "Jason's weight is above 80 Kg", the proposition "Jason's weight is 80 Kg" is necessarily true with certainty 1, while "Jason's weight is 90 Kg" is possibly true with certainty 0.5.

3 A Generic Framework for DL with Uncertainty

To incorporate uncertainty into DL, each component of the classical DL framework needs to be extended, as shown in Fig. 1. To be more specific, any framework for DL with uncertainty consists of the following three components.

1. *Description Language with Uncertainty:* The syntax and semantics of the description language are extended to enable expression of uncertainty.

2. *Knowledge Bases with Uncertainty:* A knowledge base is composed of the *intensional* knowledge, i.e., TBox (for terminological axioms) and RBox (for role axioms), both extended with uncertainty, and *extensional* knowledge, i.e., ABox (for assertions), with uncertainty.
3. *Reasoning with Uncertainty:* The DL framework is equipped with reasoning services that take into account the presence of uncertainties in DL theories during the reasoning process.

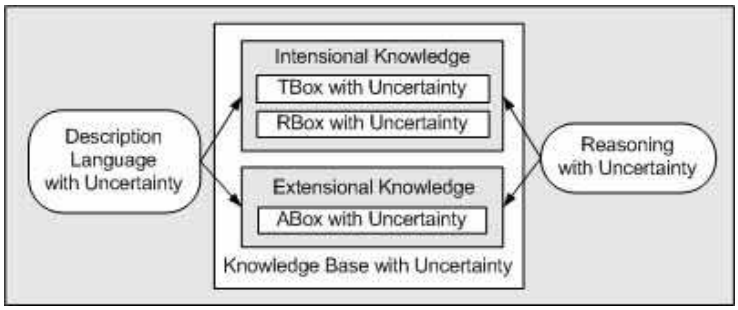

Fig. 1. DL Framework with Uncertainty

In what follows, we focus more on the first two components and discuss them in detail. The third component, the reasoning aspect, is under investigation. This section ends with some examples illustrating how different notions of uncertainty could be represented using our generic framework.

3.1 Description Language with Uncertainty

To provide a generic extension to a description language, one needs to develop a way to represent certainty values, and assign semantics to each element in the description language.

Representation of Certainty Values. To represent the certainty values, we take a *lattice-based approach* followed in the parametric framework [11]. That is, we assume that certainty values form a complete lattice shown as $\mathcal{L} = \langle \mathcal{V}, \preceq \rangle$, where $\mathcal{V}$ is the certainty domain, and $\preceq$ is the partial order defined on $\mathcal{V}$. We also use b to denote the bottom or least element in $\mathcal{V}$, and use t to denote the top or greatest value in $\mathcal{V}$. The least upper bound operator (the join operator) in $\mathcal{L}$ is denoted by $\oplus$, its greatest lower bound (the meet operator) is denoted by $\otimes$, and its negation operator is denoted by $\sim$.

The certainty lattice can be used to model both *qualitative* and *quantitative* certainty values. An example for the former is classical logic which uses the binary values $\{0, 1\}$. For the latter, an example would be a family of multi-valued logics such as fuzzy logic which uses $[0, 1]$ as the certainty domain.

Assignment of Semantics to Description Language. The generic framework treats each type of uncertainty formalism as a special case. Hence, it would be restrictive to consider any specific function to describe the semantics of the description language constructors (e.g., fixing *min* as the function to determine the certainty of concept conjunction). An alternative is proposed in our generic framework to allow a user to specify the functions that are appropriate to define the semantics of the description language element at axiom or assertion level. We elaborate more on this in section 3.2.

To make sure that the combination functions specified by a user make sense, we assume the following properties for various certainty functions to be reasonable. Most of these properties were recalled from [11], and are reasonable and justified when we verify them against existing extensions of DL with uncertainty. To present these properties, we consider the description language constructors in $\mathcal{ALC}$ [1]. We assume that the reader has the basic knowledge about $\mathcal{ALC}$.

Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be an interpretation, where $\Delta^{\mathcal{I}}$ is the domain and $\cdot^{\mathcal{I}}$ is an interpretation function that maps description language elements to some certainty value in $\mathcal{V}$.

Atomic Concept. The interpretation of an atomic concept A is a certainty value in the certainty domain, i.e., $A^{\mathcal{I}}(a) \in \mathcal{V}$, for all individuals $a \in \Delta^{\mathcal{I}}$. For example, in the fuzzy approach, the interpretation of an atomic concept A is defined as $A^{\mathcal{I}}(a) \in [0, 1]$, that is, the interpretation function assigns to every individual a in the domain, a value in the unit interval that indicates its membership to A.

Atomic Role. Similar to atomic concepts, the interpretation of an atomic role R is a certainty value in the certainty domain, i.e., $R^{\mathcal{I}}(a, b) \in \mathcal{V}$, for all individuals $a, b \in \Delta^{\mathcal{I}}$.

Top/Universal Concept. The interpretation of the top or universal concept $\top$ is the greatest value in $\mathcal{V}$, that is, $\top^{\mathcal{I}} = t$. For instance, in fuzzy, probabilistic, and possibilistic approaches, the interpretation of $\top$ is 1, or true in standard logic.

Bottom Concept. The interpretation of the bottom concept $\bot$ is the least value in $\mathcal{V}$, that is, $\bot^{\mathcal{I}} = b$. For example, in fuzzy, probabilistic, and possibilistic approaches, the interpretation of $\bot$ is 0, or false in standard logic.

Concept Negation. Given a concept C, the interpretation of concept negation $\neg C$ is defined by the negation function $\sim$, which is a mapping from $\mathcal{V}$ to $\mathcal{V}$, and should satisfy the following properties:

- Boundary Conditions: $\sim b = t$ and $\sim t = b$.
- Double Negation: For each certainty value $\alpha \in \mathcal{V}$, we have that $\sim(\sim\alpha) = \alpha$.

In our presentation here, we consider the negation operator $\sim$ in the certainty lattice as the default negation function. Other properties of the negation function, such as monotonicity (i.e., $\forall\alpha, \beta \in \mathcal{V}$, if $\alpha \preceq \beta$, then $\sim\alpha \succeq \sim\beta$), may be imposed if necessary. The most common interpretation of $\neg C$ is (1 – certainty of C), where $\mathcal{V} = [0, 1]$.

Before we present the properties of functions that are appropriate to describe the semantics of concept conjunction and disjunction, we first identify a set of desired properties which *combination function* f should satisfy. These functions are used to combine a collection of certainty values into one value. We then identify a subset of these properties suitable for describing the semantics of logical formulas on the basis of concept conjunction and disjunction. Note that, since f is used to combine a collection of certainty values into one, we describe f as a binary function from $\mathcal{V} \times \mathcal{V}$ to $\mathcal{V}$. This view is clearly without loss of generality and, at the same time, useful for implementing functions.

1. Monotonicity: $f(\alpha_1, \alpha_2) \preceq f(\beta_1, \beta_2)$, whenever $\alpha_i \preceq \beta_i$, for $i = 1, 2$.
2. Bounded Value (Above): $f(\alpha_1, \alpha_2) \preceq \alpha_i$, for $i = 1, 2$.
3. Bounded Value (Below): $f(\alpha_1, \alpha_2) \succeq \alpha_i$, for $i = 1, 2$.
4. Boundary Condition (Above): $\forall \alpha \in \mathcal{V}$, $f(\alpha, b) = \alpha$ and $f(\alpha, t) = t$.
5. Boundary Condition (Below): $\forall \alpha \in \mathcal{V}$, $f(\alpha, t) = \alpha$ and $f(\alpha, b) = b$.
6. Commutativity: $\forall \alpha, \beta \in \mathcal{V}$, $f(\alpha, \beta) = f(\beta, \alpha)$.
7. Associativity: $\forall \alpha, \beta, \delta \in \mathcal{V}$, $f(\alpha, f(\beta, \delta)) = f(f(\alpha, \beta), \delta)$.

Concept Conjunction. Given concepts C and D, the interpretation of concept conjunction $C \sqcap D$ is defined by the conjunction function f_c that should satisfy properties 1, 2, 5, 6, and 7. The monotonicity property is required so that the reasoning is monotone, i.e., whatever that has been proven so far will remain true for the rest of the reasoning process. The bounded value property is included so that the interpretation of the certainty values makes sense. Note that this property also implies the boundary condition (property 5). The commutativity property supports reordering of the arguments of the conjunction operator, and associativity ensures that different evaluation order of a conjunction of concepts does not change the result. These properties are useful during the runtime evaluation used by the reasoning procedure. Examples of conjunctions include the usual product $\times$ and *min* functions, and bounded difference defined as $bDiff(\alpha, \beta) = max(0, \alpha + \beta - 1)$.

Concept Disjunction. Given concepts C and D, the interpretation of concept disjunction $C \sqcup D$ is defined by the disjunction function f_d that should satisfy properties 1, 3, 4, 6, and 7. The monotonicity, bounded value, boundary condition, commutativity, and associativity properties are required for similar reasons as the conjunction case. Some common disjunction functions are: the standard *max* function, the probability independent function defined as $ind(\alpha, \beta) = \alpha + \beta - \alpha\beta$, and the bounded sum function defined as $bSum(\alpha, \beta) = min(1, \alpha + \beta)$.

Role Value Restriction. Given a role R and a role filler C, the interpretation of role value restriction $\forall R.C$ is defined as follows:

$\forall a \in \Delta^{\mathcal{I}}, (\forall R.C)^{\mathcal{I}}(a) = \otimes_{b \in \Delta^x} \{f_d(\sim R^{\mathcal{I}}(a, b), C^{\mathcal{I}}(b))\}$

The intuition behind this definition is to view $\forall R.C$ as the open first order formula $\forall b.\ R(a, b) \rightarrow C(b)$, where $R(a, b) \rightarrow C(b)$ is equivalent to $\neg R(a, b) \vee C(b)$,

and $\forall$ is viewed as a conjunction over the elements of the domain. To be more specific, the semantics of $\neg R(a,b)$ is captured using the negation function $\sim$ as $\sim R^{\mathcal{I}}(a,b)$, the semantics of $\neg R(a,b) \vee C(b)$ is captured using the disjunction function as $f_d(\sim R^{\mathcal{I}}(a,b), C^{\mathcal{I}}(b))$, and $\forall b$ is captured using the meet operator in the lattice $\otimes_{b \in \Delta^{\mathcal{I}}}$.

Role Exists Restriction. Given a role R and a role filler C, the interpretation of role exists restriction $\exists R.C$ is defined as follows:

$\forall a \in \Delta^{\mathcal{I}}$, $(\exists R.C)^{\mathcal{I}}(a) = \oplus_{b \in \Delta^{\mathcal{I}}}\{f_c(R^{\mathcal{I}}(a,b), C^{\mathcal{I}}(b))\}$

The intuition for this property is that we view $\exists R.C$ as the open first order formula $\exists b.\ R(a,b) \wedge C(b)$, where $\exists$ is viewed as a disjunction over the elements of the domain. To be more specific, the semantics of $R(a,b) \wedge C(b)$ is captured using the conjunction function as $f_c(R^{\mathcal{I}}(a,b), C^{\mathcal{I}}(b))$, and $\exists b$ is captured using the join operator in the lattice $\oplus_{b \in \Delta^{\mathcal{I}}}$.

Additional Inter-Constructor Properties. In addition to the above-mentioned properties, we also assume that the following inter-constructor properties hold:

- De Morgan's Rule: $\neg(C \sqcup D) \equiv \neg C \sqcap \neg D$ and $\neg(C \sqcap D) \equiv \neg C \sqcup \neg D$.
- Negating Quantifiers Rule: $\neg\exists R.C \equiv \forall R.\neg C$ and $\neg\forall R.C \equiv \exists R.\neg C$.

The above two properties are needed to convert a concept description into *negation normal form* (NNF), i.e., the negation operator appears only in front of a concept name. Note that the above properties affect the type of negation, conjunction, and disjunction functions that may be used in the generic framework.

3.2 Knowledge Bases with Uncertainty

As in the classical counterpart, a *knowledge base* Σ in the generic framework is a triple $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$, where $\mathcal{T}$ is a TBox, $\mathcal{R}$ is an RBox, and $\mathcal{A}$ is an ABox.

An interpretation $\mathcal{I}$ *satisfies* a knowledge base Σ (or $\mathcal{I} \models \Sigma$) iff it satisfies each element of Σ (i.e., $\mathcal{T}$, $\mathcal{R}$, and $\mathcal{A}$). We say that Σ is *satisfiable* (or $\Sigma \not\models \bot$) iff there exists an interpretation $\mathcal{I}$ that satisfies Σ. Similarly, Σ is *unsatisfiable* (or $\Sigma \models \bot$) iff there does not exist any interpretation $\mathcal{I}$ that satisfies Σ.

To provide a generic extension to the knowledge base, there is a need to give a syntactical and semantical extension to both the intensional (TBox and RBox) and extensional knowledge (ABox).

A TBox $\mathcal{T}$ consists of a set of terminological axioms expressed in the form $\langle C \sqsubseteq D, \alpha \rangle \langle f_c, f_d, f_p \rangle$ or $\langle C \equiv D, \alpha \rangle \langle f_c, f_d, f_p \rangle$, where C and D are concepts, $\alpha \in \mathcal{V}$ is the certainty that the axiom holds, f_c is the conjunction function used as the semantics of concept conjunction and part of the role exists restriction, f_d is the disjunction function used as the semantics of concept disjunction and part of the role value restriction, and f_p is the propagation function that is used to propagate the truth value from the LHS of the subsumption to the RHS.

Similar to the description language case, we have identified a set of properties that should hold for a propagation function. In general, the propagation

function f_p is a combination function, and should satisfy the monotonicity and bounded value (above) properties, as specified in section 3.1. Note that these two properties are a subset of those required by the conjunction functions, and they are needed for similar reasons as the conjunction case. Some commonly used propagation functions are the algebraic product $\times$ and the standard *min* function.

As usual, the concept definition $\langle C \equiv D, \alpha\rangle\langle f_c, f_d, f_p\rangle$ is defined as $\langle C \sqsubseteq D, \alpha\rangle\ \langle f_c, f_d, f_p\rangle$ and $\langle D \sqsubseteq C, \alpha\rangle\langle f_c, f_d, f_p\rangle$.

The RBox is similar to TBox except that we have role axioms instead of terminological axioms. In addition, only a propagation function is specified, but not a conjunction or disjunction functions. Since existing DL frameworks with uncertainty do not allow role conjunction or role disjunction, we do not consider them in the generic framework either.

An ABox $\mathcal{A}$ consists of a set of assertions of the form $\langle a : C, \alpha\rangle\langle f_c, f_d, -\rangle$ or $\langle (a, b) : R, \alpha\rangle\langle -, -, -\rangle$, where a and b are individuals, C is a concept, R is a role, $\alpha \in \mathcal{V}$, f_c is the conjunction function, and f_d is the disjunction function. Note that, unlike in axioms, the propagation function is not needed in the assertion, hence we use "$-$" as the placeholder to keep the uniformity of the presentation.

3.3 Knowledge Representation

In this section, we first illustrate the capabilities of the generic framework for representing classical DL. We then show how existing DLs with uncertainty can be represented by the generic framework.

*Example 1 (*Classical DL*).* The classical DL knowledge base can be represented in the generic framework as follows. The certainty lattice is defined as $\mathcal{L} = \langle \mathcal{V}, \leq\rangle$, where $\mathcal{V} = \{0, 1\}$, with max as the join operator $\oplus$ and min as the meet operator $\otimes$. Also, the negation operator is defined as $\sim 0 = 1$ and $\sim 1 = 0$. The certainty value associated with each axiom and assertion is set to 1. Finally, the conjunction function (f_c) is min, the disjunction function (f_d) is max, and the propagation function (f_p) is min. For example, consider the following classical knowledge base:

$\mathcal{T}$=$\{\langle Parent \equiv Person \sqcap \exists(hasChildByBirth.Person$
$\sqcup\ hasStepChild.Person \sqcup\ hasAdoptedChild.Person)\rangle,$
$\langle Mother \equiv Parent \sqcap Female\rangle\}$
$\mathcal{A}$=$\{\langle Mary : Person \sqcap Female\rangle,$
$\langle (Mary, Joe) : hasStepChild\rangle, \langle Joe : Person\rangle\}$

This knowledge base can be represented in the generic framework as follows:

$\mathcal{T}$=$\{\langle Parent \equiv Person \sqcap \exists(hasChildByBirth.Person \sqcup hasStepChild.$
$Person \sqcup\ hasAdoptedChild.Person),\ 1\rangle\ \langle min, max, min\rangle,$
$\langle Mother \equiv Parent \sqcap Female, 1\rangle\langle min, -, min\rangle\}$
$\mathcal{A}$=$\{\langle Mary : Person \sqcap Female, 1\rangle\langle min, -, -\rangle,$
$\langle (Mary, Joe) : hasStepChild, 1\rangle\langle -, -, -\rangle, \langle Joe : Person, 1\rangle\langle -, -, -\rangle\}$

*Example 2 (*Fuzzy DL*).* Most of the proposed fuzzy DL ('most' because our framework supports only $\mathcal{ALC}$) can be represented in the generic framework by setting the certainty lattice as $\mathcal{L} = \langle \mathcal{V}, \leq \rangle$, where $\mathcal{V} = \mathcal{C}[0,1]$ is the set of closed intervals $[\alpha, \beta]$ in [0, 1] such that $\alpha \preceq \beta$. The negation operator in this case is defined as $\sim([a_1, a_2]) = [1 - a_2, 1 - a_1]$. In [5, 13, 15, 16, 18], the meet operator is inf (infimum) and the join operator is sup (supremum). On the other hand, [17] uses min as the meet operator, and max as the join. The conjunction function used in all these proposals is min, whereas the disjunction function uses max. Note that existing proposals rarely allow certainty values to be associated with both axioms and assertions. Moreover, they do not discuss how to combine the certainty value of an assertion with the certainty value of an axiom, and hence existing frameworks do not specify any propagation function.

As an example, suppose we have the following fuzzy knowledge base:

$\mathcal{T}$={$\langle (Old \sqcup WellEducated) \sqcap LikesLearning \sqsubseteq Knowledgeable \geq 0.8 \rangle$}

$\mathcal{A}$={$\langle John : Old \geq 0.6 \rangle$, $\langle John : WellEducated \geq 0.7 \rangle$,

$\langle John : LikesLearning = 0.9 \rangle$}

Then, we could infer that John is knowledgeable with degree in [0.7, 0.9].
This knowledge base can be represented in the generic framework as follows:

$\mathcal{T}$={$\langle (Old \sqcup WellEducated) \sqcap LikesLearning \sqsubseteq Knowledgeable, [0.8, 1] \rangle$

$\langle min,\ max,\ min \rangle$}

$\mathcal{A}$={$\langle John : Old, [0.6, 1] \rangle \langle -, -, - \rangle$, $\langle John : WellEducated, [0.7, 1] \rangle \langle -, -, - \rangle$,

$\langle John : LikesLearning, [0.9, 0.9] \rangle \langle -, -, - \rangle$}

We obtain $\langle John : Knowledgeable, [0.7, 0.9] \rangle \langle -, -, - \rangle$ as the inference result.

*Example 3 (*Probabilistic DL*).* A possible certainty lattice for probabilistic DL is $\mathcal{L} = \langle \mathcal{V}, \leq \rangle$, where $\mathcal{V} = \mathcal{C}[0,1]$, with negation operator sets to be $\sim([a_1, a_2])$ $= [1 - a_2, 1 - a_1]$. Note that this allows us to express both interval probability (such as [0.4, 0.8]) and exact probability (e.g., [0.8, 0.8]).

Currently existing probabilistic extensions to DLs, such as [3, 4, 7, 8], support mainly conditional constraints. In the generic framework, we view a rule as a conditional statement. As such, let α be some value from the certainty lattice, a conditional constraint of the form $P(A \mid B) : \alpha$ can be expressed as the rule $A \overset{\alpha}{\leftarrow} B$ in IB framework of logic programming, which in turn can be expressed as $\langle B \sqsubseteq A, \alpha \rangle$ in our generic framework.

For illustration purpose, we show how a simple knowledge base with conditional constraints from [4] can be represented in our generic framework. Consider a knowledge base with the following expressions:

$\mathcal{T}$={$\langle PacemakerPatient \sqsubseteq HeartPatient \rangle$,

$\langle (\exists hasPrivateInsurance.\{Yes\} | HeartPatient)[0.9, 1] \rangle$}

$\mathcal{A}$={$\langle (PacemakerPatient | \{John\})[0.8, 1] \rangle$}

Then, one could infer that {$\langle (\exists hasPrivateInsurance.\{Yes\} | \{John\})[0.72, 1] \rangle$}
The above knowledge base can be expressed in the generic framework as follows:

$\mathcal{T}$={$\langle PacemakerPatient \sqsubseteq HeartPatient, [1, 1] \rangle \langle -, -, \times \rangle$,

$\langle HeartPatient \sqsubseteq \exists hasPrivateInsurance.\{Yes\}, [0.9, 1] \rangle \langle -, -, \times \rangle$}

$\mathcal{A}$={$\langle John : PacemakerPatient, [0.8, 1] \rangle \langle -, -, - \rangle$}

With the first axiom and the first assertion, we infer that John is HeartPatient with probability $\times([0.8,1],[1,1]) = [0.8,1]$ (since f_p in the first axiom is $\times$). Then, this assertion together with the second axiom allow us to infer that John has private insurance with probability $\times([0.8,1],[0.9,1]) = [0.72,1]$ (since the f_p in the second axiom is $\times$), as inferred in [4].

Unlike in other uncertainty formalisms, reasoning with probability requires extra information/knowledge. Hence, although our framework can easily handle simple probabilities, such as independent events and mutually exclusive events, more complex probability modes such as positive/negative correlation [9], ignorance [9], and conditional independence [14] are still under investigation.

*Example 4 (*Possibilistic DL*).* Hollunder [6] is the only proposal that gives a possibilistic extension to DL. Here, the possibility (Π) and necessity (N) degrees can be represented by the certainty lattice $\mathcal{L} = \langle \mathcal{V}, \leq \rangle$, where $\mathcal{V} = \mathcal{C}[0,1]$, with negation operator sets to be $\sim([a_1,a_2]) = [1-a_2, 1-a_1]$. The conjunction and propagation functions are min, and the disjunction function is max.
As an example, in [6], we have the following knowledge base:

$\mathcal{T}$={$\langle \exists owns.Porsche \sqsubseteq RichPerson \sqcup CarFanatic \geq N0.8 \rangle$,
$\langle RichPerson \sqsubseteq Golfer \geq \Pi 0.7 \rangle$}
$\mathcal{A}$={$\langle Tom : \exists owns.Porsche \geq N1 \rangle$, $\langle Tom : \neg CarFanatic \geq N0.7 \rangle$}

Then, according to [6], one could infer that $\langle Tom : Golfer \geq \Pi 0.7 \rangle$.
The above knowledge base can be simulated in the generic framework as follows:

$\mathcal{T}$={$\langle \exists owns.Porsche \sqsubseteq RichPerson \sqcup CarFanatic, [0.8,1] \rangle$
$\langle min, max, min \rangle$, $\langle RichPerson \sqsubseteq Golfer, [0.7,1] \rangle \langle -,-,min \rangle$}
$\mathcal{A}$={$\langle Tom : \exists owns.Porsche, [1,1] \rangle \langle min,-,- \rangle$,
$\langle Tom : \neg CarFanatic, [0.7,1] \rangle \langle -,-,- \rangle$}

The result of the inference is $\langle Tom : Golfer, [0.7,1] \rangle \langle -,-,- \rangle$, as obtained in [6].

4 Conclusion and Future Work

We have proposed a generic framework that allows us to unify and/or generalize various extensions of DLs with uncertainty, taking an axiomatic approach. In particular, we abstracted away both the underlying notion of uncertainty (fuzzy logic, probability, possibilistic logic), and the way in which the constructors in the description language are interpreted. This was done by identifying the essential properties that various combination functions should possess in order to realize our unifying framework. We are currently investigating ways to specify generic reasoning services that the proposed framework should support, for which we have some partial result. Other future work includes extending this framework to a more expressive DL, for instance $\mathcal{SHOIN}$. A study of suitable query processing and optimization techniques is an important future work. An implementation of the proposed generic framework is underway.

Acknowledgements: This work is supported in part by Natural Sciences and Engineering Research Council (NSERC) of Canada, and by ENCS, Concordia University.

References

1. Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F., editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
2. Berners-Lee, T., Hendler, J., and Lassila, O. The Semantic Web. *Scientific American*, 284(5), May 2001.
3. Ding, Z., Peng, Y., and Pan, R. A Bayesian approach to uncertainty modeling in OWL ontology. In *Proceedings of AISTA-04*, Luxembourg, November 2004.
4. Giugno, R. and Lukasiewicz, T. P-$\mathcal{SHOQ}$(D): A probabilistic extension of $\mathcal{SHOQ}$(D) for probabilistic ontologies in the Semantic Web. In *Proceedings of the JELIA-02*, pages 86–97, London, UK, 2002. Springer-Verlag. Lecture Notes In Computer Science; Vol. 2424.
5. Hölldobler, S., Khang, T. D., and Störr, H.-P. A fuzzy description logic with hedges as concept modifiers. In *Proceedings of InTech/VJFuzzy-02*, pages 25–34, Hanoi, Vietnam, 2002. Science and Technics Publishing House.
6. Hollunder, B. An alternative proof method for possibilistic logic and its application to terminological logics. In *Proceedings of UAI-94*, pages 327–335, San Francisco, CA, 1994. Morgan Kaufmann.
7. Jaeger, M. Probabilistic reasoning in terminological logics. In *Proceedings of KR-94*, pages 305–316, 1994.
8. Koller, D., Levy, A. Y., and Pfeffer, A. P-CLASSIC: A tractable probablistic description logic. In *Proceedings of AAAI-97*, pages 390–397, Providence, Rhode Island, July 1997. AAAI Press.
9. Lakshmanan, L.V.S. and Sadri, F. Probabilistic deductive databases. In *Proceedings of ILPS-94*, pages 254–268, Ithaca, NY, November 1994. MIT Press.
10. Lakshmanan, L.V.S. and Shiri, N. Logic programming and deductive databases with uncertainty: A survey. In *Enclyclopedia of Computer Science and Technology*, volume 45, pages 153–176. Marcel Dekker, Inc., New York, 2001.
11. Lakshmanan, L.V.S. and Shiri, N. A parametric approach to deductive databases with uncertainty. *IEEE Transactions on Knowledge and Data Engineering*, 13(4):554–570, 2001.
12. Parsons, S. Current approaches to handling imperfect information in data and knowledge bases. *IEEE Transactions on Knowledge and Data Engineering*, 8(3):353–372, 1996.
13. Sánchez, D. and Tettamanzi, A. G. B. Generalizing quantification in fuzzy description logics. In *Proceedings of Fuzzy Days-04*. Springer-Verlag, 2004.
14. Shenoy P.P. Conditional independence in valuation-based systems. *Journal of Approximate Reasoning*, 10(3):203–234, 1994.
15. Straccia, U. Reasoning within fuzzy description logics. *Journal of Artificial Intelligence Research*, 14:137–166, 2001.
16. Straccia, U. Transforming fuzzy description logics into classical description logics. In *Proceedings of JELIA-04*, pages 385–399. Springer-Verlag, 2004. Lecture Notes In Computer Science; Vol. 3229.
17. Stracia, U. A fuzzy description logic. In *Proceedings of AAAI-98*, pages 594–599, Menlo Park, CA, USA, 1998. AAAI Press.
18. Tresp, C. and Molitor, R. A description logic for vague knowledge. In *Proceedings of ECAI-98*, pages 361–365, Brighton, UK, 1998. John Wiley and Sons.
19. Zadeh, L. A. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
20. Zadeh, L. A. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1):3–28, 1978.