Description Logics over Lattices with Multi-valued Ontologies

Stefan Borgwardt and **Rafael Peñaloza**
TU Dresden, Germany
{penaloza,stefborg}@tcs.inf.tu-dresden.de

Abstract

Uncertainty is unavoidable when modeling most application domains. In medicine, for example, symptoms (such as pain, dizziness, or nausea) are always subjective, and hence imprecise and incomparable. Additionally, concepts and their relationships may be inexpressible in a crisp, clear-cut manner. We extend the description logic $\mathcal{ALC}$ with multi-valued semantics based on lattices that can handle uncertainty on concepts as well as on the axioms of the ontology. We introduce reasoning methods for this logic w.r.t. general concept inclusions and show that the complexity of reasoning is not increased by this new semantics.

1 Introduction

Description logics (DLs) [Baader *et al.*, 2003] are a family of logic-based knowledge representation formalisms, which have been employed in various application domains and whose most notable success so far is the adoption of the DL-based language OWL as a standard for the semantic web. However, in their usual form, DLs lack the ability to handle uncertainty. Uncertainty is unavoidable in areas such as medicine, where determining without doubt whether a patient has a specific disease may require intrusive or expensive tests. For example, a full diagnosis for anemia requires a blood test measuring the number of red blood cells and hemoglobin level. However, this test should only be performed if there is some valid reason to suspect an abnormality in these levels. Additionally, when preparing a diagnosis, experts must deal with symptoms reported by the patients, which are by definition subjective, and hence imprecise and incomparable.

Moreover, the relationship between diseases and their external manifestations is rarely clear-cut. For instance, the anemic syndrome may be caused by a very wide range of maladies (undernourishment, leukemia, drepanocytosis, etc.) which may be more or less likely depending on the ethnic origin, gender, or activity of the patient, but none of these factors is fully determinant to the origin of the syndrome.

We propose an extension of classical DLs that can handle uncertainty through multi-valued semantics. In our approach we consider a finite number of truth values, organized in a lattice extended with a negation operator. The semantics of this logic is a standard generalization from classical DLs in the sense that concepts (classically interpreted as sets) are now interpreted as multi-valued sets over the background lattice. The semantics of the concept constructors is also generalized from Boolean to lattice operators.

A previous approach combining DLs with multi-valued semantics based on lattices was presented in [Straccia, 2006], but can only deal with acyclic terminologies containing crisp axioms. Thus, it is unable to express cyclic relations necessary for representing, e.g. hereditary diseases: a patient having a relative with drepanocytosis is likely, but not certain, to exhibit this disorder as well. In this paper we develop reasoning procedures that are able to handle general concept inclusion axioms, which may themselves include a degree of uncertainty. We can then reason over cyclic axioms like[1]

$$\langle \exists \mathsf{relative}.\mathsf{Drepanocytosis} \sqsubseteq \mathsf{Drepanocytosis}, \mathsf{likely} \rangle\,,$$

which expresses the fact that drepanocytosis is a hereditary disease, i.e. a patient having a relative with drepanocytosis is likely, but not certain, to exhibit this disorder as well.

In order to present the main ideas of our approach, we restrict ourselves to $\mathcal{ALC}_L$, the multi-valued variant of $\mathcal{ALC}$ over the lattice L. It should be clear, however, that the same ideas can be transferred to more expressive DLs. We then describe an exponential-time procedure for reasoning in this logic, which is ExpTime-hard.

In Section 2 we present some basic notions of lattice theory and the logic $\mathcal{ALC}_L$ with its main reasoning problems. We then show how to reason in this logic on the assumption that it has the *witnessed model property*. In Section 4 we prove that the witnessed model property may not hold in general, but the algorithm, with minor modifications, remains correct. Finally, in Section 5, we give an overview of the wide range of approaches that have been proposed to deal with uncertainty and imprecision in DL, and compare them to our own. Due to a lack of space we only sketch the proofs of our results.

2 A Multi-valued Description Logic

In this section we first introduce some basic notions of lattice theory and then define the lattice-based multi-valued description logic $\mathcal{ALC}_L$.

[1] The semantics of this expression can be found in Definition 3.

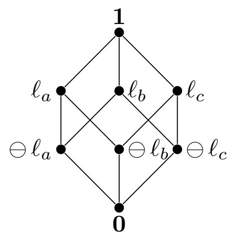

Figure 1: The De Morgan lattice L_3.

2.1 Lattices

A *lattice* is an algebraic structure $(L, \oplus, \otimes)$ consisting of a *carrier* set L and two binary operations *supremum* $\oplus$ and *infimum* $\otimes$ that are commutative and associative, and satisfy the absorption laws $\ell \oplus (\ell \otimes m) = \ell = \ell \otimes (\ell \oplus m)$ for all $\ell, m \in L$.[2] The order $\leq$ on L is defined by $\ell \leq m$ iff $\ell \otimes m = \ell$ for all $\ell, m \in L$. A lattice is called *distributive* if $\oplus$ and $\otimes$ distribute over each other, *finite* if L is finite, and *bounded* if it has a *minimum* and a *maximum* element, denoted as $\mathbf{0}$ and $\mathbf{1}$, respectively. Every finite lattice is also bounded. In a finite lattice, we may use the notation $\bigoplus_{t \in T} t$ ($\bigotimes_{t \in T} t$) for the supremum (infimum) of a set $T \subseteq L$. Whenever it is clear from the context, we will simply use the carrier set L to represent the lattice $(L, \oplus, \otimes)$.

A *De Morgan lattice* is a bounded distributive lattice extended with an involutive unary operation $\ominus$, called *negation*, that satisfies De Morgan's laws $\ominus(\ell \oplus m) = \ominus\,\ell \otimes \ominus\, m$ and $\ominus(\ell \otimes m) = \ominus\,\ell \oplus \ominus\, m$ for all $\ell, m \in L$. The negation is an anti-monotone bijection on the lattice. Figure 1 shows a simple De Morgan lattice.

The operators of a De Morgan lattice can be seen as the natural generalization of the logical operators $\vee, \wedge$, and $\neg$. Based on this intuition, we define the *implication* of two elements in the lattice as $\ell \Rightarrow m := \ominus\,\ell \oplus m$.

For the rest of this paper, we assume that L is an arbitrary, but fixed, finite De Morgan lattice. The elements of this lattice will describe the certainty of assertions. For instance, one could use values in a total order (e.g. *unlikely*, *likely*, *very likely*, *sure*) or use several dimensions to express incomparable values of uncertainty (e.g. subjective measurements made by different sources).

2.2 Multi-valued $\mathcal{ALC}$

The multi-valued description logic $\mathcal{ALC}_L$ is a generalization of the crisp DL $\mathcal{ALC}$ that allows the use of the elements of a De Morgan lattice as truth values, instead of just the Boolean values *true* and *false*. The syntax of concept descriptions in $\mathcal{ALC}_L$ is the same as in $\mathcal{ALC}$.

Definition 1 (syntax of $\mathcal{ALC}_L$)**.** Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be two disjoint sets of *concept names* and *role names*, respectively. $\mathcal{ALC}_L$ *concept descriptions* are built through the following syntactic rule:

$$C ::= A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid \exists r.C \mid \forall r.C \mid \top \mid \bot$$

where $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and C, C_1, C_2 are $\mathcal{ALC}_L$ concept descriptions.

The semantics of this logic is given by an interpretation function that not simply describes whether an element of the domain belongs to a concept, but gives a lattice value describing the certainty with which the element satisfies this concept; in other words, the semantics is based on multi-valued sets.

Definition 2 (semantics of $\mathcal{ALC}_L$)**.** An *interpretation* is a pair $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ where $\Delta^\mathcal{I}$ is a non-empty *domain* and $\cdot^\mathcal{I}$ is a function that assigns to every concept name A and every role name r functions $A^\mathcal{I} : \Delta^\mathcal{I} \to L$ and $r^\mathcal{I} : \Delta^\mathcal{I} \times \Delta^\mathcal{I} \to L$, respectively. The function $\cdot^\mathcal{I}$ is extended to $\mathcal{ALC}_L$ concept descriptions as follows:

- $(C \sqcap D)^\mathcal{I}(x) = C^\mathcal{I}(x) \otimes D^\mathcal{I}(x)$,
- $(C \sqcup D)^\mathcal{I}(x) = C^\mathcal{I}(x) \oplus D^\mathcal{I}(x)$,
- $(\neg C)^\mathcal{I}(x) = \ominus\, C^\mathcal{I}(x)$,
- $(\exists r.C)^\mathcal{I}(x) = \bigoplus_{y \in \Delta^\mathcal{I}} r^\mathcal{I}(x, y) \otimes C^\mathcal{I}(y)$,
- $(\forall r.C)^\mathcal{I}(x) = \bigotimes_{y \in \Delta^\mathcal{I}} r^\mathcal{I}(x, y) \Rightarrow C^\mathcal{I}(y)$,
- $\top^\mathcal{I}(x) = \mathbf{1}, \bot^\mathcal{I}(x) = \mathbf{0}$.

for every $x \in \Delta^\mathcal{I}$.

Notice that the existential and universal quantifiers are dual, i.e. $\neg\exists r.C$ and $\forall r.\neg C$ have the same semantics for every $\mathcal{ALC}_L$ concept description C and every role name r.

The knowledge of a domain is usually stored in an *ontology*, which is a collection of axioms. In this paper, we restrict to terminological knowledge, given by a so-called TBox.

Definition 3 (TBox)**.** A *TBox* is a finite set of *(labeled) general concept inclusions* (GCIs) of the form $\langle C \sqsubseteq D, \ell \rangle$, where C, D are $\mathcal{ALC}_L$ concept descriptions and $\ell \in L$.

An interpretation $\mathcal{I}$ is a *model* of the TBox $\mathcal{T}$ if it satisfies all axioms in $\mathcal{T}$, i.e. if for every axiom $\langle C \sqsubseteq D, \ell \rangle \in \mathcal{T}$ it holds that $\bigotimes_{x \in \Delta^\mathcal{I}} C^\mathcal{I}(x) \Rightarrow D^\mathcal{I}(x) \geq \ell$.

We emphasize here that $\mathcal{ALC}$ is a special case of $\mathcal{ALC}_L$, where the underlying lattice contains only the elements $\mathbf{0}$ and $\mathbf{1}$, which may be interpreted as *false* and *true*, respectively. Accordingly, one can think of generalizing the reasoning problems for $\mathcal{ALC}$ to the use of other lattices. The standard reasoning problems for crisp DLs are satisfiability and subsumption of concepts. In our setting, we are further interested in the degree of certainty with which these properties hold.

Definition 4 (satisfiability, subsumption)**.** Let C, D be $\mathcal{ALC}_L$ concept descriptions, $\mathcal{T}$ a TBox and $\ell \in L$. C is *ℓ-satisfiable* w.r.t. $\mathcal{T}$ if there is a model $\mathcal{I}$ of $\mathcal{T}$ with $\bigoplus_{x \in \Delta^\mathcal{I}} C^\mathcal{I}(x) \geq \ell$. The *best satisfiability degree* for C w.r.t. $\mathcal{T}$ is the largest ℓ such that C is ℓ-satisfiable w.r.t. $\mathcal{T}$.

C is *ℓ-subsumed* by D w.r.t. $\mathcal{T}$ if every model $\mathcal{I}$ of $\mathcal{T}$ is also a model of $\langle C \sqsubseteq D, \ell \rangle$. The *best subsumption degree* for C and D is the largest $\ell \in L$ such that C is ℓ-subsumed by D w.r.t. $\mathcal{T}$.

Notice that if C is ℓ-satisfiable and ℓ'-satisfiable w.r.t. $\mathcal{T}$, then C is also $\ell \oplus \ell'$-satisfiable. Likewise for ℓ-subsumption. Hence, the notions of best satisfiability and best subsumption degrees are well defined. Moreover, as the following lemma

[2] See [Grätzer, 1998] for a more detailed introduction to lattices.

shows, it is sufficient to develop an algorithm for finding the best satisfiability degree of a concept.

Lemma 5. *Let C, D be two concept descriptions, $\mathcal{T}$ a TBox, and $\ell \in L$. The best satisfiability degree for $C \sqcap \neg D$ is ℓ iff the best subsumption degree for C and D is $\ominus \ell$.*

The best satisfiability degree of a concept is important for medical applications, where a doctor may want to find out how likely a given pathology is. For example, the best satisfiability degree of $\mathsf{Male} \sqcap \mathsf{Hepatomegalia} \sqcap \mathsf{Anemia}$ expresses the likelihood of finding an hepatomegalic male with anemia. In some cases, however, this notion of satisfiability turns out to be too weak, since a concept C may be ℓ-satisfiable even if no element of the domain may ever belong to C with a value greater or equal to ℓ.

Example 6. We use the lattice L_3 from Figure 1. The concept A is $\mathbf{1}$-satisfiable w.r.t. the TBox $\mathcal{T}$ having the axioms

$$\langle A \sqsubseteq \neg B, \ell_b \rangle, \quad \langle A \sqsubseteq B, \mathbf{1} \rangle$$

since $\mathcal{I}_0 = (\{x_1, x_2\}, \cdot^{\mathcal{I}_0})$ with $A^{\mathcal{I}_0}(x_1) = B^{\mathcal{I}_0}(x_2) = \ell_a$, $B^{\mathcal{I}_0}(x_1) = A^{\mathcal{I}_0}(x_2) = \ell_c$ is a model of $\mathcal{T}$ and $\ell_a \oplus \ell_c = \mathbf{1}$. However, there is no model $\mathcal{I}$ of $\mathcal{T}$ where $A^{\mathcal{I}}(x) = \mathbf{1}$ for some $x \in \Delta^{\mathcal{I}}$. If such $\mathcal{I}$ and x existed, then the first axiom would restrict $B^{\mathcal{I}}(x)$ to be $\leq \ominus \ell_b$ (either $\ominus \ell_b$ or $\mathbf{0}$), while the second axiom would force this value to be $\mathbf{1}$.

For this reason, we consider a stronger notion of satisfiability that requires at least one element of the domain to satisfy the concept with the given value. A concept C is *strongly ℓ-satisfiable* w.r.t. a TBox $\mathcal{T}$ if there is a model $\mathcal{I}$ of $\mathcal{T}$ and an $x \in \Delta^{\mathcal{I}}$ such that $C^{\mathcal{I}}(x) \geq \ell$. Obviously, strong ℓ-satisfiability implies ℓ-satisfiability. As shown in Example 6, the converse does not hold. However, satisfiability can be reduced to strong satisfiability by means of the following lemma.

Lemma 7. *The best satisfiability degree for C w.r.t. $\mathcal{T}$ is the supremum of all ℓ such that C is strongly ℓ-satisfiable.*

Proof Sketch. If C is strongly ℓ- and strongly ℓ'-satisfiable, there exist two models $\mathcal{I}, \mathcal{I}'$ of $\mathcal{T}$ and $x \in \Delta$, $x' \in \Delta'$ with $C^{\mathcal{I}}(x) \geq \ell$ and $C^{\mathcal{I}'}(x') \geq \ell'$. The disjoint union of $\mathcal{I}$ and $\mathcal{I}'$ gives a model $\mathcal{J}$ where $\bigoplus_{y \in \Delta^{\mathcal{J}}} C^{\mathcal{J}}(y) \geq \ell \oplus \ell'$. □

We can then find out whether C is ℓ-satisfiable by comparing ℓ to the best satisfiability degree of C. We will thus focus on developing an algorithm for finding all the lattice elements that witness the strong satisfiability of a given concept. As we will show, this reasoning problem is not harder than deciding satisfiability of crisp $\mathcal{ALC}$ concepts.

3 Deciding Strong Satisfiability

We now present an automata-based algorithm for deciding the strong satisfiability of a concept. To simplify the construction, we first consider reasoning over witnessed models only (see Definition 8). We later show (in Section 4) that this restriction is not necessary for the correctness of the algorithm.

Definition 8 (witnessed model)**.** Let $\eta \in \mathbb{N}$. A model $\mathcal{I}$ of a TBox $\mathcal{T}$ is called *η-witnessed* if for every $x \in \Delta^{\mathcal{I}}$ and every concept description of the form $\exists r.C$ there are η elements $x_1, \dots, x_\eta \in \Delta^{\mathcal{I}}$ such that

$$(\exists r.C)^{\mathcal{I}}(x) = \bigoplus_{i=1}^{\eta} r^{\mathcal{I}}(x, x_i) \otimes C^{\mathcal{I}}(x_i),$$

and analogously for the universal restrictions $\forall r.C$. In particular, if $\eta = 1$, then the suprema and infima from the semantics of $\exists r.C$ and $\forall r.C$ become maxima and minima, respectively. In this case, we simply say that $\mathcal{I}$ is *witnessed*.

We will present a procedure to check strong satisfiability w.r.t. witnessed models which is based on the emptiness check of finite automata working on infinite trees. But first, we give a brief introduction to this kind of automata.

As input structure we consider the infinite k-ary tree K^* for $K := \{1, \dots, k\}$ with $k \in \mathbb{N}$. The positions of the *nodes* in this tree are represented through words in K^* in the usual way: the empty word ε represents the root node, and ui represents the i-th successor of the node u.

Definition 9 (looping automaton)**.** A *looping automaton (LA)* is a tuple (Q, I, Δ) consisting of a finite set Q of *states*, a set $I \subseteq Q$ of *initial states* and a *transition relation* $\Delta \subseteq Q \times Q^k$.

A *run* of this automaton is a mapping $r : K^* \to Q$ that assigns states to each node of K^* such that (i) $r(\varepsilon) \in I$ and (ii) for every $u \in K^*$, $(r(u), r(u1), \dots, r(uk)) \in \Delta$.

The *emptiness problem* for LA is to decide whether a given LA has a run.

The emptiness problem for LA can be solved by the following procedure in polynomial time. The idea is to incrementally build the set of all states that cannot appear in any run; we will call these *bad states*. All states without transitions are clearly bad states, and hence the set is initialized with those states. On each iteration, we add to this set all states that only have transitions leading to bad states. This set becomes stable after at most $|Q|$ iterations. The automaton has a run iff there is an initial state that is not bad. It is worth to point out that, as a side-effect, this procedure computes the set of *all* non-bad initial states without additional effort.

We now return to the problem of deciding strong ℓ-satisfiability of $\mathcal{ALC}_L$ concept descriptions. Our automata-based approach relies on the fact that a concept is strongly ℓ-satisfiable iff it has a well-structured tree model, called a *Hintikka tree*. Intuitively, Hintikka trees are abstract representations of tree models, that express the membership value of all "relevant" concept descriptions. The automaton will have exactly these Hintikka trees as its runs. Strong ℓ-satisfiability is thus reduced to the emptiness test of an automaton.

In the following we assume that all concept descriptions are in *negation normal form* (NNF); that is, negation appears only in front of concept names. Any $\mathcal{ALC}_L$ concept description can be transformed to NNF using the De Morgan rules, duality of quantifiers, and elimination of double negations. We denote the NNF of C by $\mathsf{nnf}(C)$ and $\mathsf{nnf}(\neg C)$ by $\backsim C$. The concept description $\mathsf{nnf}(C)$ always has the same semantics as C, since we are using a De Morgan lattice and existential and universal restrictions are dual to each other.

We denote as $\mathsf{sub}(C, \mathcal{T})$ the set of all subconcepts of C and of the concept descriptions $\backsim D \sqcup E$ for $\langle D \sqsubseteq E, \ell \rangle \in \mathcal{T}$.

The states of the automaton will be so-called Hintikka sets. Strictly speaking, these are multi-valued sets: every element has an associated membership value from the lattice L. Their domain is the set $\mathsf{sub}(C, \mathcal{T})$, together with an arbitrary element ρ.

Definition 10 (Hintikka set)**.** A function $H : \mathsf{sub}(C, \mathcal{T}) \cup \{\rho\} \to L$ is called a *(multi-valued) Hintikka set* for $C, \mathcal{T}$ if it satisfies the conditions (i) $H(D \sqcap E) = H(D) \otimes H(E)$, (ii) $H(D \sqcup E) = H(D) \oplus H(E)$, and (iii) for every concept name A, $H(\neg A) = \ominus H(A)$.

The Hintikka set H is *compatible* with the GCI $\langle D \sqsubseteq E, \ell \rangle$ if $H(\backsim D \sqcup E) \geq \ell$.

The arity k of our automaton is determined by the number of existential and universal restrictions, i.e. concept descriptions of the form $\exists r.D$ or $\forall r.D$, contained in $\mathsf{sub}(C, \mathcal{T})$. Intuitively, each successor acts as the witness for one of these restrictions. The additional domain element ρ is used to express the degree with which the role relation to the parent node holds. To know which successor in the tree corresponds to which restriction, we fix an arbitrary bijection $\varphi : \{E \mid E \in \mathsf{sub}(C, \mathcal{T}) \text{ is of the form } \exists r.D \text{ or } \forall r.D\} \to K$. The following Hintikka conditions define the transitions of our automaton.

Definition 11 (Hintikka condition)**.** The tuple of Hintikka sets $(H_0, H_1, \ldots, H_k)$ for $C, \mathcal{T}$ satisfies the *Hintikka condition* if: (i) for every existential restriction $\exists r.D$, $H_0(\exists r.D) = H_{\varphi(\exists r.D)}(\rho) \otimes H_{\varphi(\exists r.D)}(D)$ and additionally for every restriction F of the form $\exists r.E$ or $\forall r.E$, it holds $H_0(\exists r.D) \geq H_{\varphi(F)}(\rho) \otimes H_{\varphi(F)}(D)$; and (ii) for every universal restriction $\forall r.D$, $H_0(\forall r.D) = H_{\varphi(\forall r.D)}(\rho) \Rightarrow H_{\varphi(\forall r.D)}(D)$ and additionally for every restriction F of the form $\exists r.E$ or $\forall r.E$, it holds $H_0(\forall r.D) \leq H_{\varphi(F)}(\rho) \Rightarrow H_{\varphi(F)}(D)$.

A *Hintikka tree* for $C, \mathcal{T}$ is an infinite k-ary tree $\mathbf{T}$ labeled with Hintikka sets that are compatible with every GCI in $\mathcal{T}$ where, for every node $u \in K^*$, the tuple $(\mathbf{T}(u), \mathbf{T}(u1), \ldots, \mathbf{T}(uk))$ satisfies the Hintikka condition. The compatibility condition ensures that all axioms are satisfied at any node of the Hintikka tree, while the Hintikka condition makes sure that the tree is in fact a witnessed model.

Recall that for now we are only considering reasoning *w.r.t. witnessed models*, namely, decide whether there is a witnessed model $\mathcal{I}$ such that $C^{\mathcal{I}}(x) \geq \ell$ for some $x \in \Delta^{\mathcal{I}}$. The proof of the following theorem uses arguments similar to those in [Baader *et al.*, 2008].

Theorem 12. *Let C be a concept description and $\mathcal{T}$ a TBox. Then C is strongly ℓ-satisfiable w.r.t. $\mathcal{T}$ (in a witnessed model) iff there is a Hintikka tree $\mathbf{T}$ for $C, \mathcal{T}$ such that $\mathbf{T}(\varepsilon)(C) \geq \ell$.*

Proof Sketch. A Hintikka tree can be seen as a witnessed model with domain K^* and interpretation function given by the Hintikka sets. Thus, any Hintikka tree $\mathbf{T}$ for $C, \mathcal{T}$ with $\mathbf{T}(\varepsilon)(C) \geq \ell$ entails strong ℓ-satisfiability of C w.r.t. $\mathcal{T}$.

On the other hand, every witnessed model $\mathcal{I}$ with a domain element $x \in \Delta^{\mathcal{I}}$ for which $C^{\mathcal{I}}(x) \geq \ell$ holds can be *unraveled* into a Hintikka tree $\mathbf{T}$ for $C, \mathcal{T}$ as follows. We start by labeling the root node by the Hintikka set that records the membership values of x for each concept from $\mathsf{sub}(C, \mathcal{T})$. We then create successors of the root by considering every element of $\mathsf{sub}(C, \mathcal{T})$ of the form $\exists r.D$ or $\forall r.D$ and finding the witness $y \in \Delta^{\mathcal{I}}$ for this restriction. We create a new node for y which is an r-successor of the root node with degree $r^{\mathcal{I}}(x, y)$. By continuing this process, we construct a Hintikka tree $\mathbf{T}$ for $C, \mathcal{T}$ for which $\mathbf{T}(\varepsilon)(C) \geq \ell$ holds. ☐

Thus, in order to decide strong ℓ-satisfiability w.r.t. witnessed models, we only need to decide emptiness of the following automaton.

Definition 13 (Hintikka automaton)**.** Let C be an $\mathcal{ALC}_L$ concept description, $\mathcal{T}$ a TBox, and $\ell \in L$. The *Hintikka automaton* for $C, \mathcal{T}, \ell$ is the LA $\mathcal{A}_{C,\mathcal{T},\ell} = (Q, I, \Delta)$ where Q is the set of all compatible Hintikka sets for $C, \mathcal{T}$, I contains all Hintikka sets H with $H(C) \geq \ell$, and Δ is the set of all $(k + 1)$-tuples of compatible Hintikka sets that satisfy the Hintikka condition.

The runs of $\mathcal{A}_{C,\mathcal{T},\ell}$ are exactly the Hintikka trees $\mathbf{T}$ having $\mathbf{T}(\varepsilon)(C) \geq \ell$. Thus, C is strongly ℓ-satisfiable w.r.t. $\mathcal{T}$ iff $\mathcal{A}_{C,\mathcal{T},\ell}$ is non-empty. Since the automaton $\mathcal{A}_{C,\mathcal{T},\ell}$ is exponential in $C, \mathcal{T}$, and the emptiness test for looping automata is polynomial in the size of the automaton, overall we obtain an exponential time decision procedure for strong ℓ-satisfiability. This bound is optimal, because concept satisfiability is already EXPTIME-hard for crisp $\mathcal{ALC}$ with general concept inclusions [Baader *et al.*, 2003].

Theorem 14. *The problem of deciding strong ℓ-satisfiability (w.r.t. witnessed models) of an $\mathcal{ALC}_L$ concept description C w.r.t. a TBox $\mathcal{T}$ is* EXPTIME*-complete.*

Furthermore, the emptiness test described before can be used to compute the set of *all* Hintikka sets that may appear at the root of a Hintikka tree. From this we can extract the set of all values ℓ such that $\mathbf{T}(\epsilon)(C) \geq \ell$ for some Hintikka tree $\mathbf{T}$. From Lemma 7 it then follows that the best satisfiability degree for C w.r.t. $\mathcal{T}$ can also be computed in exponential time. By Lemma 5, all of the reasoning problems defined in this paper are EXPTIME-complete.

We emphasize here that this complexity analysis does not consider the underlying lattice L as part of the input. This is a reasonable assumption as, for any given application, the lattice will never be modified, and hence all the lattice-based operations can be hardcoded to be performed in constant time. However, if the size of the lattice is measured as the number of its elements, then the algorithm is also exponential in this value, since all the lattice operations can be performed in polynomial time in this size.

4 Dealing With More Witnesses

The use of the Hintikka automata from Definition 13 for deciding strong ℓ-satisfiability is only correct if $\mathcal{ALC}_L$ has the *witnessed model property*, i.e. every strongly ℓ-satisfiable concept is also strongly ℓ-satisfiable w.r.t. witnessed models. However, this property does not hold in general.

Example 15. Consider the lattice L_3 and the TBox $\mathcal{T}$ from Example 6. If we extend the interpretation $\mathcal{I}_0$ from the same example such that $r^{\mathcal{I}_0}(x_1, x_1) = r^{\mathcal{I}_0}(x_1, x_2) = \mathbf{1}$, then we see that $\exists r.A$ is strongly $\mathbf{1}$-satisfiable. However, in any witnessed model $\mathcal{I}$, the strong $\mathbf{1}$-satisfiability of this concept would imply the existence of an individual x such that $A^{\mathcal{I}}(x) = \mathbf{1}$, which was shown to be impossible in Example 6.

Nonetheless, there is always a constant $\eta \in \mathbb{N}$, depending only on the underlying lattice L, such that $\mathcal{ALC}_L$ has the *η-witnessed model property*; i.e. strong ℓ-satisfiability is equivalent to strong ℓ-satisfiability w.r.t. η-witnessed models. The number η depends on the *compactness degree* of L.

Definition 16 (compactness)**.** Let $\eta \in \mathbb{N}$. A lattice L is called *η-compact* if for every $A \subseteq L$ there is a subset $B \subseteq A$ with at most η elements such that $\bigoplus B = \bigoplus A$. The *compactness degree* of L is the smallest η for which L is η-compact.

The following lemma states that the compactness degree yields a bound on the number of successors that witness any existential and universal restriction. It is an easy consequence of Definitions 8 and 16.

Lemma 17. *Let L be a finite De Morgan lattice and $\eta \in \mathbb{N}$. If L is η-compact, then $\mathcal{ALC}_L$ has the η-witnessed model property.*

Moreover, the compactness degree of a given lattice L is bounded by the *width* of L, i.e. the cardinality of the longest antichain of L.

Lemma 18. *If L has width $\eta \in \mathbb{N}$, then L is η-compact.*

Proof. If a set $A \subseteq L$ contains two comparable elements, then we can always remove one of them without affecting the supremum of A. □

This implies that $\mathcal{ALC}_L$ always has the η-witnessed model property for some $\eta \in \mathbb{N}$, since we assumed L to be finite. From Lemma 18 we deduce that every total order is 1-compact. Consequently, if L is a finite total order with De Morgan negation, then $\mathcal{ALC}_L$ has the witnessed model property. This is the case, e.g. for fuzzy $\mathcal{ALC}$ based on the Zadeh fuzzy operations [Straccia, 2001].

The constructions of the previous section can easily be adapted for η-witnessed models. We can introduce the notion of *η-witnessed Hintikka trees* and construct looping automata $\mathcal{A}_{\eta,C,\mathcal{T},\ell}$ such that the following are equivalent: (i) C is strongly ℓ-satisfiable w.r.t. $\mathcal{T}$ (in an η-witnessed model), (ii) there is an η-witnessed Hintikka tree $\mathbf{T}$ for $C, \mathcal{T}$ such that $\mathbf{T}(\varepsilon)(C) \geq \ell$, and (iii) the LA $\mathcal{A}_{\eta,C,\mathcal{T},\ell}$ is non-empty. The main difference lies in the arity k of the Hintikka trees. Thus, a similar algorithm can be applied even if the logic does not have the witnessed model property.

Theorem 19. *The problem of deciding strong ℓ-satisfiability of an $\mathcal{ALC}_L$ concept description C w.r.t. a TBox $\mathcal{T}$ is* EXPTIME*-complete.*

As before, the complexity of this problem does not change if we do not view the lattice L as fixed, but measure its size as the number of its elements.

5 Related Work

Several different formalisms have so far been suggested for dealing with uncertainty or vagueness in DL ontologies.

In [Straccia, 2006], a similar approach to the one presented here was taken. The description logic $\mathcal{ALC}$ is augmented by multi-valued interpretations over a De Morgan lattice. The difference to our approach lies in the treatment of axioms. In [Straccia, 2006], the TBox axiom $C \sqsubseteq D$ is satisfied by an interpretation $\mathcal{I}$ if $C^{\mathcal{I}}(x) \leq D^{\mathcal{I}}(x)$ holds for all $x \in \Delta^{\mathcal{I}}$. Additionally, only acyclic TBoxes are allowed, and thus, all defined concepts can be expanded beforehand. This eliminates the need to deal with the TBox in the algorithm that checks satisfiability. Consequently, the presented tableaux algorithm only checks ABox consistency in $\mathcal{ALC}_L$ w.r.t. an empty TBox. Our approach using automata allows us to deal with arbitrary TBox axioms expressing, e.g. hereditary diseases; on the other hand, we do not consider ABox axioms. [Jiang *et al.*, 2010] extend the work of [Straccia, 2006] to the more expressive DL $\mathcal{SHIN}$. However, they limit the expressivity of the terminological axioms in the same way.

Fuzzy DLs are another way to deal with uncertainty in ontologies [Yen, 1991; Lukasiewicz and Straccia, 2008]. These formalisms apply the ideas of fuzzy sets [Zadeh, 1965] to description logics. Concepts are interpreted as mappings from the domain into the unit interval $[0, 1]$. Depending on the underlying logical operators, the logics differ in the interpretation of the concept constructors. However, the total order $[0, 1]$ does not allow for incomparable degrees of uncertainty as our approach does.

Our definition of $\mathcal{ALC}_L$ includes the fuzzy DL with Zadeh semantics and Kleene-Dienes-implication [Straccia, 2001; Stoilos *et al.*, 2007]. In this case, the semantics is as that of $\mathcal{ALC}_L$ for the De Morgan lattice $L = ([0, 1], \max, \min)$ with negation $x \mapsto 1 - x$. Although this lattice is not finite, once the TBox is fixed we can restrict ourselves w.l.o.g. to a finite subset of $[0, 1]$, as $\max$ and $\min$ create no new values. The tableau algorithm that was developed for reasoning in fuzzy $\mathcal{ALC}$ with Zadeh semantics and general TBoxes generalizes the well-known algorithm for crisp $\mathcal{ALC}$ and may thus require non-deterministic exponential time in the worst case. Our algorithm improves on this by giving an EXPTIME upper bound, thus proving EXPTIME-completeness of the problem for the first time. Since our approach satisfies desirable properties such as idempotency of conjunction and duality of quantifiers, it is unable to treat more complex t-norm based semantics [Bobillo and Straccia, 2009a].

A different way to enrich DLs with uncertainty is to follow the approach of rough set theory [Pawlak, 1982]. The basic idea of rough DLs [Schlobach *et al.*, 2007; Keet, 2010] is that some concepts cannot be described precisely, but only by some *lower* and *upper approximation*. The semantics of this logic requires, in addition to the interpretation of the different concepts, an equivalence relation, called *indiscernibility*, on the elements of the domain, which is used to formalize these approximations. This formalism assumes that one can specify a global indiscernibility relation on the domain, whereas the idea behind multi-valued DLs is that one is able to locally specify a membership degree for each individual. The authors show that the rough set semantics can be translated to classical DL. This is predicated on the existence of transitivity, symmetry and reflexivity axioms for roles, which do not exist in $\mathcal{ALC}$. In any DL that includes these axioms, one can easily translate the indiscernibility equivalence relation of rough set theory into a special role without adding to the complexity of reasoning. However, for $\mathcal{ALC}$ this reduction yields a strict increase in complexity. Efforts have also been made to combine the approaches of rough set theory and fuzzy logic into a single description logic [Bobillo and Straccia, 2009b].

6 Conclusions

We presented a general framework for reasoning under uncertainty based on the description logic $\mathcal{ALC}$. We considered the reasoning tasks of deciding ℓ-satisfiability and ℓ-subsumption as well as computing the best satisfiability and subsumption degrees w.r.t. general TBoxes. It turns out that these reasoning tasks are in the same complexity class as for the crisp DL $\mathcal{ALC}$. In [Straccia, 2006] it was shown that satisfiability w.r.t. crisp acyclic TBoxes is PSPACE-complete.

To our knowledge, this is the first time that automata-based techniques were used to analyze reasoning in DLs augmented by De Morgan lattices. This enabled us to treat the case of general concept inclusions instead of the simpler acyclic TBoxes. Although we did not consider reasoning w.r.t. ABoxes, it is clear that a preprocessing method similar to the one presented in [Hollunder, 1996], combining the tableaux algorithm developed in [Straccia, 2006] with this automata-based approach, would allow for the simultaneous treatment of ABoxes and general TBoxes. We also plan to extend our approach to more expressive DLs, as was already done for reasoning w.r.t. crisp TBoxes in [Jiang *et al.*, 2010]. Considering the close relationship between the automaton deciding satisfiability in $\mathcal{ALC}$ and our approach, we believe that a similar argument can be used to show that multi-valued $\mathcal{SI}$ is PSPACE-complete with acyclic TBoxes and EXPTIME-complete with general TBoxes [Baader *et al.*, 2008].

It would also be interesting to see whether concept modifiers (e.g. `(very)Tall`) can be treated in this framework. Additionally, one may consider an implication different from the one we used here, e.g. the residuum of the infimum $\otimes$.

The advantage of description logics over more expressive logical formalism has always been the feasibility for implementation. Regarding the treatment of uncertainty, there are several possible ways to implement reasoning procedures. One could implement a completely new system or augment an existing system like RACER[3] or FaCT++[4] to deal with multi-valued instead of crisp semantics.

Furthermore, it may be possible to reduce reasoning in $\mathcal{ALC}_L$ to reasoning in $\mathcal{ALC}$. This has already been done for fuzzy DLs [Straccia, 2004] and multi-valued DLs based on linear orders [Straccia, 2006]. The advantage of this approach is that one can reuse existing optimized reasoners for crisp DLs. However, it remains to be seen whether this is feasible for arbitrary De Morgan lattices. Finally, one could also use a straightforward translation of $\mathcal{ALC}_L$ into a multi-valued first order language and use existing reasoners for these more expressive formalisms, e.g. $_3T^AP$.[5]

[3] `http://www.racer-systems.com/`

[4] `http://owl.man.ac.uk/factplusplus`

[5] `http://i12www.ira.uka.de/threetap`

References

[Baader *et al.*, 2003] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. CUP, 2003.

[Baader *et al.*, 2008] F. Baader, J. Hladik, and R. Peñaloza. Automata can show PSPACE results for description logics. *Inf. and Comput.*, 206(9–10):1045–1056, 2008.

[Bobillo and Straccia, 2009a] F. Bobillo and U. Straccia. Fuzzy description logics with general t-norms and datatypes. *Fuzzy Sets and Systems*, 160(23):3382–3402, 2009.

[Bobillo and Straccia, 2009b] F. Bobillo and U. Straccia. Supporting fuzzy rough sets in fuzzy description logics. In *Proc. of ECSQARU'09*, volume 5590 of *LNCS*, pages 676–687. Springer, 2009.

[Grätzer, 1998] G. Grätzer. *General Lattice Theory*. Birkhäuser, Basel, second edition edition, 1998.

[Hollunder, 1996] B. Hollunder. Consistency checking reduced to satisfiability of concepts in terminological systems. 18(2–4):133–157, 1996.

[Jiang *et al.*, 2010] Y. Jiang, Y. Tang, J. Wang, P. Deng, and S. Tang. Expressive fuzzy description logics over lattices. *Knowledge-Based Systems*, 23(2):150–161, 2010.

[Keet, 2010] C. M. Keet. Ontology engineering with rough concepts and instances. In *Proc. of EKAW 2010*, volume 6317 of *LNCS*, pages 503–513. Springer, 2010.

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

[Pawlak, 1982] Z. Pawlak. Rough sets. *Int. J. of Computer and Information Sciences*, 11/5:341–356, 1982.

[Schlobach *et al.*, 2007] S. Schlobach, M. C. A. Klein, and L. Peelen. Description logics with approximate definitions: Precise modeling of vague concepts. In *Proc. of IJCAI'07*, Hyderabad, India, January 2007.

[Stoilos *et al.*, 2007] G. Stoilos, G. B. Stamou, J. Z. Pan, V. Tzouvaras, and I. Horrocks. Reasoning with very expressive fuzzy description logics. *JAIR*, 30:273–320, 2007.

[Straccia, 2001] U. Straccia. Reasoning within fuzzy description logics. *JAIR*, 14:137–166, 2001.

[Straccia, 2004] U. Straccia. Transforming fuzzy description logics into classical description logics. In *Proc. of JELIA'04*, volume 3229 of *LNCS*, pages 385–399. Springer, 2004.

[Straccia, 2006] U. Straccia. Description logics over lattices. *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, 14(1):1–16, 2006.

[Yen, 1991] J. Yen. Generalizing term subsumption languages to fuzzy logic. In *Proc. of IJCAI'91*, pages 472–477, 1991.

[Zadeh, 1965] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, June 1965.