SEMANTIC TECHNOLOGY INSTITUTE (STI)

ON FIXPOINT-BASED DECISION PROCEDURES FOR FUZZY DESCRIPTION LOGICS I

Uwe Keller Stijn Heymans

STI TECHNICAL REPORT 2008-08-07

OCTOBER 16, 2008

SEMANTIC TECHNOLOGY INSTITUTE (STI)

STI Innsbruck
University of Innsbruck
Technikerstrasse 21a
Innsbruck, Austria
www.sti-innsbruck.at

STI TECHNICAL REPORT

STI TECHNICAL REPORT 2008-08-07, OCTOBER 16, 2008

ON FIXPOINT-BASED DECISION PROCEDURES FOR FUZZY DESCRIPTION LOGICS I

Uwe Keller[1] Stijn Heymans[2]

Abstract. We present **FixIt**($\mathbb{ALC}$), a novel procedure for deciding knowledge base (KB) satisfiability in the Fuzzy Description Logic (FDL) $\mathbb{ALC}$. **FixIt**($\mathbb{ALC}$) does not search for tree-structured models as in tableau-based proof procedures, but embodies a (greatest) fixpoint-computation of canonical models that are not necessarily tree-structured, based on a type-elimination process. Soundness, completeness and termination are proven and the runtime and space complexity are discussed. We give a precise characterization of the worst-case complexity of deciding KB satisfiability (as well as related terminological and assertional reasoning tasks) in $\mathbb{ALC}$ in the general case and show that our method yields a worst-case optimal decision procedure (under reasonable assumptions). To the best of our knowledge it is the first *fixpoint-based* decision procedure for FDLs, hence introducing a new class of inference procedures into FDL reasoning.

Keywords: Fuzzy Description Logics, Fuzzy Modal Logics, Knowledge-based Satisfiability, Type Elimination, Fixpoint-Computation, Canonical Models, Symbolic Representation, BDDs

[1]Semantic Technology Institute (STI) Innsbruck, University of Innsbruck, Technikerstraße 21a, A-6020 Innsbruck, Austria. E-mail: uwe.keller@sti-innsbruck.at

[2]Knowledge Based Systems Group, Institute of Information Systems, Vienna University of Technology, Favoritenstrasse 9-11, A-1040 Vienna, Austria. E-mail: heymans@kr.tuwien.ac.at

Acknowledgements: This work has been partially funded by the European Commission under the LarKC project (FP7 - 215535). Stijn Heymans is supported by the Austrian Science Fund (FWF) under projects P20305 and P20840.

Contents

1 Introduction

Description Logics (DLs) [2] are a popular family of formally well-founded and decidable knowledge representation languages. Generally speaking they can be understood as the modal logics of information systems. DLs have a wide range of applications, but are most widely known as the basis for ontology languages used in the context of the Semantic Web (SW) such as the Web Ontology Language OWL [15]. Fuzzy Description Logics (FDLs) [21] extend DLs to represent *vague* concepts and relations. Uncertainty of this form naturally arises in many practical applications of knowledge-based systems, in particular the SW. FDLs for instance fit very well to the problem of multimedia information retrieval [13]. Another feature that makes FDLs specifically interesting for the SW is a basic form of para-consistency, i.e. a statement and its negation are possible to hold at the same time (to a certain extent). This allows knowledge providers on the SW to disagree on the basic properties of data object and their interrelation without causing the (uninformative) explosion of the deductive closure as in classical DLs.

So far, reasoning in Fuzzy DLs is mainly based on tableau-methods (e.g. [21, 19, 11, 20, 24, 7]). Further, [22] demonstrates how to use inference procedures for classical DLs to perform reasoning in (some) FDLs. Still, reasoning in FDLs is at least as hard as reasoning in classical (crisp) DLs. Even in DLs of modest expressivity (e.g. $\mathbb{ALC}$ [21, 22, 19] the fuzzy variant of the DL $\mathcal{ALC}$ [18]) the worst-case complexity of reasoning is significant (cf. Section 3.1) even in restricted cases [21]. Therefore, it is clear that there can not be a *single* inference method that works well on *all* problems.

Consequently, our goal is to enrich the range of available methods for reasoning with FDLs with a fundamentally different approach. In practical applications of DLs (and hence FDLs) a particularly important feature for representing domain models is the support of so-called *general terminologies* (see e.g. [19]), i.e., the possibility to capture (potentially recursive) interdependencies between complex concepts in a domain model. However, besides the tableau-based methods for DLs (e.g [19, 11, 24, 7]) there are at present no other FDL inference methods which can deal with general terminologies. We want to provide an alternative to tableau-based methods that can deal with general terminologies.

Contributions. The main contributions of the paper are as follows:

- We present a novel procedure $\mathbf{FixIt}(\mathbb{ALC})$ (cf. Section 5.2) for deciding knowledge base (KB) satisfiability in the FDL $\mathbb{ALC}$ (cf. Section 2).
- We clarify the worst-case complexity of the reasoning task addressed by our algorithm and show formally that the problem is EXPTIME-complete. From this result, we can further establish EXPTIME-completeness for a range of related terminological and assertional reasoning tasks (cf. Section 3.1).
- We formally prove soundness, completeness and termination of the algorithm (cf. Section 5.2) and show that the runtime behavior of the proposed algorithm is worst-case optimal (cf. Section 5.4).
- $\mathbf{FixIt}(\mathbb{ALC})$ generalizes a type-elimination-based decision procedure [16] for the (classical) modal logic $\mathbf{K}$ (i.e. $\mathcal{KBDD}$ [14]) to the FDL $\mathbb{ALC}$. Additionally we integrate (fuzzy) ABoxes and general TBoxes which are not dealt with in $\mathcal{KBDD}$.
- To the best of our knowledge it is the first *fixpoint-based* decision procedure that has been proposed for FDL introducing a *new class of inference procedures* into FDL reasoning.
- Besides the tableau-based methods in [19, 11, 24, 7], it is the only approach to integrate general terminologies in FDL reasoning and the first non-tableau-based one that we are aware of. General

terminologies are handled in a fundamentally different way than in standard tableau-based method such as [19, 11].

Our method is interesting especially regarding the last aspect since the handling of TBoxes in standard tableau-based methods (e.g. [19, 11]) is the *major* source of non-determinism (cnf. Section 4) and hence computational inefficiencies in implementations. In our case no non-deterministic choice is introduced by terminologies at all.

Overview of the Paper. This paper is structured further as follows: Section 2 summarizes the main definitions and results needed throughout the paper. Section 3 explores the complexity of the most common reasoning tasks in the context of general KBs in $\mathbb{ALC}$, in particular for KB satisfiability, which is the reasoning task that is addressed by our novel decision procedure $\mathbf{FixIt}(\mathbb{ALC})$. Section 4 summarizes why the proposed method might be interesting especially in regard of reasoning with GCIs. The novel decision procedure is then discusses in detail 5. Section 6 overviews relevant related work. Section 7 concludes and presents future research activities.

2 Preliminaries

We summarize the main definitions and results needed as prerequisites for presenting our approach and relating it to previous work. Section 2.1 introduces a basic classical DL, which is later extended to allow for the representation of *vague* concepts in Section 2.2, i.e. to the fuzzy DL $\mathbb{ALC}$. Finally, we overview reasoning in $\mathbb{ALC}$ in Section 2.3.

2.1 The Description Logic $\mathcal{ALC}$

Description Logics are a family of *class-based* knowledge representation formalisms characterized by the use of various constructors to build complex classes from simpler ones, and by an emphasis on the provision of sound, complete and (empirically) tractable reasoning services.

We introduce a prototypical DL called $\mathcal{ALC}$ [18, 2, 3]. Although $\mathcal{ALC}$ is a simple DL, its fundamental reasoning tasks have significant worst-case complexity. DLs used in applications today (e.g. $\mathcal{SHIQ}$, $\mathcal{SHOIN}$, or $\mathcal{ALB}$) often extend $\mathcal{ALC}$. Further, $\mathcal{ALC}$ is implemented in various inference systems based on different algorithms.

In the following, we summarize the syntax, semantics and inference problems of $\mathcal{ALC}$ and give the main complexity-theoretic as well as model-theoretic properties.

Syntax. Let $\mathbf{C}$ denote a non-empty, countable set of concept names, $\mathbf{R}$ denote a countable set of role names such that $\mathbf{C}$, $\mathbf{R}$ are pairwise disjoint. Then, we call $\Sigma = (\mathbf{C}, \mathbf{R}, \mathbf{I})$ a *signature*.

The set $\mathcal{C}(\Sigma)$ of *concepts over* Σ is the smallest set of expressions that contains $\mathbf{C}$ and is closed under concept union (denoted by $C \sqcup D$), concept complement (denoted by $\neg C$), and existential role restriction (denoted by $\exists R.C$) for any $C, D \in \mathcal{C}(\Sigma)$ and $R \in \mathbf{R}$. We consider the expression $C \sqcap D$ (called concept intersection) as an abbreviation for $\neg(\neg C \sqcup \neg D)$. We use $\top$ as an abbreviation for the concept expressions $C \sqcup \neg C$ (for some fixed $C \in \mathbf{C}$) denoting the *top concept*. The *bottom concept* $\bot$ abbreviates $\neg\top$. Finally, we define universal role restriction (denoted by $\forall R.C$) as the abbreviation for $\neg\exists R.\neg C$.

A *terminology* $\mathcal{T}$ is a finite set of general concept inclusion axioms $\mathcal{T} = \{C_1 \sqsubseteq D_1, \ldots, C_n \sqsubseteq D_n\}$ where $C_i, D_i \in \mathcal{C}(\Sigma)$. We consider a concept equality axiom $C \equiv D$ as an abbreviation for the two inclusion axioms $C \sqsubseteq D$ and $D \sqsubseteq C$.

DLs can be equipped with a model-theoretic semantics. The concept of and interpretation of a signature is introduced to assign meaning to the description language. As usual, interpretations are extended to complex descriptions by the definition of the semantics of each constructor in the language, i.e. concept and role constructors in the case of DLs.

Semantics. Let $\Sigma = (\mathbf{C}, \mathbf{R}, \mathbf{I})$ be a signature. A Σ-*interpretation* $\mathrm{I} = (\Delta^\mathrm{I}, \cdot^\mathrm{I})$ consists of a non-empty set Δ^I, called the *domain* of I, and an evaluation function $\cdot^\mathrm{I}$ which maps each concept name $C \in \mathbf{C}$ to a subset $C^\mathrm{I} \subseteq \Delta^\mathrm{I}$ of the domain and each role name $R \in \mathbf{R}$ to a subset $R^\mathrm{I} \subseteq \Delta^\mathrm{I} \times \Delta^\mathrm{I}$. The evaluation function $\cdot^\mathrm{I}$ of I is extended from concept names and role names to concepts as shown in Figure1.

An interpretation I *satisfies a concept inclusion axiom* $C \sqsubseteq D$ (denoted by $\mathrm{I} \models C \sqsubseteq D$) iff $C^\mathrm{I} \subseteq D^\mathrm{I}$. I *satisfies the terminology* $\mathcal{T}$ (or is a *model of* $\mathcal{T}$, denoted by $\mathrm{I} \models \mathcal{T}$) iff I satisfies all $C \sqsubseteq D \in \mathcal{T}$.

Constructor	*Syntax* E	*Semantics* E^I (wrt. a (crisp) interpretation I)
concept conjunction	$C \sqcap D$	$C^\mathrm{I} \cap D^\mathrm{I}$
concept disjunction	$C \sqcup D$	$C^\mathrm{I} \cup D^\mathrm{I}$
concept negation	$\neg C$	$\Delta^\mathrm{I} \setminus C^\mathrm{I}$
exist. value restriction	$\exists R.C$	$\{x \in \Delta^\mathrm{I} \mid \exists y.(x,y) \in R^\mathrm{I} \wedge y \in C^\mathrm{I}\}$
univ. value restriction	$\forall R.C$	$\{x \in \Delta^\mathrm{I} \mid \forall y.(x,y) \in R^\mathrm{I} \rightarrow y \in C^\mathrm{I}\}$

Figure 1: Semantics of Constructors in $\mathcal{ALC}$.

A very fundamental inference problem commonly studied for DLs (and in fact implemented in most DL deduction systems) is to decide if a given concept is satisfiable, i.e. if it is possible for the concept to have instances. One can consider this question in a more general context by taking background knowledge into account, in particular a terminology. This specific inference problem in fact can provide a basis for solving other inference problems (e.g. concept subsumption) by means of polynomial-time reductions.

Definition 2.1 (Concept Satisfiability)**.** *Let* $\Sigma = (\mathbf{C}, \mathbf{R}, \mathbf{I})$ *be a signature,* $\mathcal{T}$ *be a terminology over* Σ *and* $C \in \mathcal{C}(\Sigma)$ *be a concept.* C *is* satisfiable *iff there exists a* Σ*-interpretation* $\mathrm{I} = (\Delta^\mathrm{I}, \cdot^\mathrm{I})$ *such that* $C^\mathrm{I} \neq \emptyset$*.* C *is* satisfiable wrt. the terminology $\mathcal{T}$ *iff there exists a* Σ*-interpretation* I *such that* $C^\mathrm{I} \neq \emptyset$ *and* $\mathrm{I} \models \mathcal{T}$*.*

Complexity of Reasoning. Deciding concept satisfiability wrt. a (general) terminology is an EXPTIME-complete problem [17, 6]. It becomes a PSPACE-complete problem if background terminologies are not considered at all (i.e. $\mathcal{T} = \emptyset$) or restricted to only contain definitions of concept names without cyclic dependencies [18, 1]. $\mathcal{ALC}$ enjoys the tree model property [4], i.e if a concept C is satisfiable wrt. $\mathcal{T}$, then there exists a model $\mathrm{I} = (\Delta^\mathrm{I}, \cdot^\mathrm{I})$ of $\mathcal{T}$ such that $\bigcup_{R \in \mathbf{R}} R^\mathrm{I}$ forms a tree over Δ^I, and $C^\mathrm{I} \neq \emptyset$. The tree must not be finite, if we allow background terminologies. In fact, there are cases where all tree models are infinite, e.g. for any concept C with the terminology $\mathcal{T} = \{C \sqsubseteq \exists R.C\}$. In the restricted cases of $\mathcal{T} = \emptyset$ or an acyclic $\mathcal{T}$, $\mathcal{ALC}$ enjoys even the *finite* tree model property, i.e. satisfiable concepts C have a tree model I such that Δ^I is a finite set.

Relation to Modal Logics. It has been shown in [17] that the DL $\mathcal{ALC}$ is a syntactic variant of the (multi-)modal logic $\mathbf{K}_{(\mathbf{m})}$. In this context, a concept expression corresponds to modal formula and a (set-theoretic)

interpretations as presented above corresponds formally to a Kripke-structure. The concept satisfiability problem essentially is the same as the problem to determine the satisfiability of a formula in $\mathbf{K}_{(\mathbf{m})}$. Finally, it is easy to see that the concept satisfiability problem wrt. to terminologies has strong similarities with correspondence theory for modal logics [4], i.e. the terminology restricts the class of models that are considered during the reasoning process.

2.2 The Fuzzy Description Logic $\mathbb{ALC}$

We introduce $\mathbb{ALC}$ [21], the fuzzy variant of the Description Logic $\mathcal{ALC}$ introduced in Section 2.1. $\mathbb{ALC}$ provides the starting point for more expressive FDLs [23] that have been proposed to fuzzify major fragments of the Web Ontology Language OWL [15].

Syntax. The set of concept expressions is defined exactly as in $\mathcal{ALC}$: expressions are constructed from an initial signature $\Sigma = (\mathbf{C}, \mathbf{R}, \mathbf{I})$ consisting of concept names $\mathbf{C}$, role names $\mathbf{R}$ and individual names $\mathbf{I}$. The set of concept expressions $\mathcal{C}(\Sigma)$ over Σ is defines as the smallest set of expressions that contains $\mathbf{C}$, $\top$ and is closed under the application of the concept constructors $C \sqcap D$ (concept intersection), $C \sqcup D$ (concept union), $\neg C$ (concept complement), and $\forall R.C$ (universal role restriction) for all $R \in \mathbf{R}$ and $C, D \in \mathcal{C}(\Sigma)$. We additionally allow expressions of the form $\exists R.C$ for $C \in \mathcal{C}(\Sigma), R \in \mathbf{R}$ and $\bot$ and consider them as syntactic shortcuts for expressions of the form $\neg\forall R.\neg C$ and $\neg\top$ respectively. A TBox axiom (or general concept inclusion axiom (GCI)) is an expression of the form $C \sqsubseteq D$ s.t. $C, D \in \mathcal{C}(\Sigma)$. A terminology (or TBox) $\mathcal{T}$ is a finite set of TBox axioms. Syntactically, the vagueness of descriptions becomes explicit only when describing specific instances and their interrelations: A (fuzzy) ABox axiom is an expression of form $\langle i : C \bowtie d\rangle$ or $\langle R(i, i') \geq d\rangle$ s.t. $i, i' \in \mathbf{I}$, $d \in [0, 1]$ and $\bowtie \in \{\leq, \geq, =\}$. An ABox $\mathcal{A}$ is a finite set of ABox axioms. Finally, a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consists of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. Let $Ind_{\mathcal{A}} \subseteq \mathbf{I}$ denote the individual names that occur in $\mathcal{A}$.

Semantics. Meaning in $\mathbb{ALC}$ is captured by means of a model-theoretic semantics. Semantically, vagueness is hereby reflected in the use of fuzzy sets and relations when interpreting concepts and roles: An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty (crisp) set $\Delta^{\mathcal{I}}$ called the domain, and a function $\cdot^{\mathcal{I}}$ which maps each concept name $C \in \mathbf{C}$ to a fuzzy set $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0, 1]$, each role name $R \in \mathbf{R}$ to a fuzzy relation $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0, 1]$ and each individual names $i \in \mathbf{I}$ to an element $i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The interpretation function $\cdot^{\mathcal{I}}$ is extended to arbitrary concept expressions $C \in \mathcal{C}(\Sigma)$ as in shown in Fig.2.

Constructor	**Syntax** E	**Semantics** $E^{\mathcal{I}}(o)$ (wrt. interpretation $\mathcal{I}$)
concept conjunction	$C \sqcap D$	$min(C^{\mathcal{I}}(o), D^{\mathcal{I}}(o))$
concept disjunction	$C \sqcup D$	$max(C^{\mathcal{I}}(o), D^{\mathcal{I}}(o))$
concept negation	$\neg C$	$1 - C^{\mathcal{I}}(o)$
exist. value restriction	$\exists R.C$	$inf_{o' \in \Delta^{\mathcal{I}}}\{max(1 - R^{\mathcal{I}}(o, o'), C^{\mathcal{I}}(o'))\}$
universal truth / possibility	$\top$	1

Figure 2: Semantics of Constructors in $\mathbb{ALC}$.

Remark 2.1 (Definability of $\top$ and $\bot$)**.** *In contrast to classical DLs it is not possible to define the semantics of $\top$ and $\bot$ by means of a concept expression build from the atomic concept names using concept intersec-*

tion, concept negation, concept union, universal role restriction only. In particular, it does not *hold that* $(C \sqcup \neg C)^{\mathcal{I}} = \top^{\mathcal{I}}$ *for all interpretations* $\mathcal{I}$*. Hence, we need to add* $\top$ *(or* $\bot$*) to the language explicitly.*

An interpretation $\mathcal{I}$ satisfies a TBox axiom $\alpha = C \sqsubseteq D$ iff for all $o \in \Delta^{\mathcal{I}}$ it holds that $C^{\mathcal{I}}(o) \leq D^{\mathcal{I}}(o)$,i.e. if C is a fuzzy subset of D. $\mathcal{I}$ satisfies an ABox axiom $\alpha = \langle i : C \bowtie d \rangle$ iff $C^{\mathcal{I}}(i^{\mathcal{I}}) \bowtie d$. $\mathcal{I}$ satisfies an ABox axiom $\alpha = \langle R(i, i') \geq d \rangle$ iff $R^{\mathcal{I}}(i^{\mathcal{I}}, i'^{\mathcal{I}}) \geq d$. In all these cases, we write $\mathcal{I} \models \alpha$. $\mathcal{I}$ satisfies a TBox $\mathcal{T}$ (or is a model of $\mathcal{T}$) iff $\mathcal{I} \models \alpha$ for all $\alpha \in \mathcal{T}$. $\mathcal{I}$ satisfies an ABox $\mathcal{A}$ (or is a model of $\mathcal{A}$) iff $\mathcal{I} \models \alpha$ for all $\alpha \in \mathcal{A}$. Finally, $\mathcal{I}$ satisfies a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ (or is a model of $\mathcal{K}$) iff $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$.

2.3 Reasoning in $\mathbb{ALC}$

Given a fuzzy KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, fuzzy ABox axioms or GCIs α and concept expressions $C, D \in \mathcal{C}(\Sigma)$, we can analyze particular semantic characteristics and interdependencies: We say that $\mathcal{K}$ is *satisfiable* (or consistent) iff there is a model $\mathcal{I}$ for $\mathcal{K}$. $\mathcal{K}$ *entails* α (denoted as $\mathcal{K} \models \alpha$) iff all models $\mathcal{I}$ of $\mathcal{K}$ satisfy α. Concept C is subsumed by concept D (wrt. a KB $\mathcal{K}$) iff $\mathcal{K} \models C \sqsubseteq D$. Two concepts C and D are called *equivalent* (wrt. a KB $\mathcal{K}$) iff for any model $\mathcal{I}$ of $\mathcal{K}$ it holds that $C^{\mathcal{I}}(o) = D^{\mathcal{I}}(o)$ for all $o \in \Delta^{\mathcal{I}}$. Two concepts C and D are called *disjoint* (wrt. a KB $\mathcal{K}$) iff for any model $\mathcal{I}$ of $\mathcal{K}$ it holds that there does not exists an $o \in \Delta^{\mathcal{I}}$ such that $C^{\mathcal{I}}(o) > 0$ and $D^{\mathcal{I}}(o) > 0$. A concept C is called satisfiable (wrt. a KB $\mathcal{K}$) iff there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}}(o) > 0$ for some $o \in \Delta^{\mathcal{I}}$. Further, one might want to compute the truth value bounds for a given ABox assertion α wrt. $\mathcal{K}$ to determine the possibility interval that is enforced for α by the background knowledge in $\mathcal{K}$: The *greatest lower bound* of α wrt. $\mathcal{K}$ is defined as $glb(\alpha, \mathcal{K}) := sup\{d \,|\, \mathcal{K} \models \langle \alpha \geq d \rangle\}$ and the *least upper bound* of α wrt. $\mathcal{K}$ is defined as $lub(\alpha, \mathcal{K}) := inf\{d \,|\, \mathcal{K} \models \langle \alpha \leq d \rangle\}$ (where $sup\, \emptyset = 0$ and $inf\, \emptyset = 1$). Computing $glb(\alpha, \mathcal{K})$ and $lub(\alpha, \mathcal{K})$ is usually called the *best truth value bounds* (BTVB) problem.

One of the most fundamental reasoning problems is to determine whether a given fuzzy KB $\mathcal{K}$ is satisfiable. A lot of other reasoning tasks (e.g., checking for concept satisfiability wrt. a TBox or the BTVB problem) can be reduced to KB satisfiability checking [21] and therefore solved by a respective decision procedure. For this reason, we consider KB satisfiability as the reasoning problem to be solved in the remaining part of the paper.

3 Complexity of Reasoning with Knowledge Bases

Surprisingly little research has been done on the complexity of reasoning in FDLs. In particular for the basic FDL $\mathbb{ALC}$ considered here, the only complexity related results are given in [21], whereby the formalism considered there is restricted in the sense that GCIs are not allowed.

In this section, we study systematically the complexity of various reasoning tasks for the FDL $\mathbb{ALC}$ as presented in Section 2.2: Section 3.1 considers the problem of deciding KB satisfiability. Section 3.2 discusses terminological reasoning. Finally, extensional reasoning is considered in Section3.3.

3.1 Satisfiability of Knowledge Bases

Deciding the satisfiability of KBs in $\mathbb{ALC}$ where the TBox $\mathcal{T}$ is restricted to axioms of the form $A \sqsubseteq C$ or $A \equiv C$ (for concept names $A \in \mathbf{C}$ and concept expressions $C \in \mathcal{C}(\Sigma)$) such that any concept name A occurs at most once on the left-hand side and the TBox does not contain any cyclic dependencies between concept

names is known to be a PSPACE-complete problem [21]. For the general case of unrestricted terminologies (allowing arbitrary GCIs $C \sqsubseteq D$), we are not aware of any worst-case characterization of the runtime complexity. We therefore show in this section that determining the satisfiability of a KB in the general case is EXPTIME-complete (which corresponds to the situation in the classical variant $\mathcal{ALC}$).

ExpTime-Hardness. We show ExpTime-hardness by a polynomial-time reduction of concept satisfiability wrt. general terminologies in the classical DL $\mathcal{ALC}$ which is known to be an ExpTime-hard problem[1] [2, Chapter 3]:

We define the necessary reduction function as follows:

Definition 3.1 (Reduction)**.** *Let* T *denote a finite set of GCIs and* $C \in \mathcal{C}(\Sigma)$ *be any concept expression. Then we define the reduction* $\pi(\mathrm{T}, C)$ *of the terminology* T *and the concept expression* C *as* $\pi(\mathrm{T}, C) := \mathcal{K}$ *where* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *is the* $\mathbb{ALC}$ *knowledge base consisting of the TBox* $\mathcal{T} := \mathcal{T}_1 \cup \mathcal{T}_2$ *with*

$$\mathcal{T}_1 = \{\top \sqsubseteq \neg D \sqcup D' \mid D \sqsubseteq D' \in \mathrm{T}\}$$
$$\mathcal{T}_2 := \{\top \sqsubseteq \neg D \sqcup D \mid D \in \mathsf{sub}(\mathrm{T} \cup \{C\})\}$$

and the ABox $\mathcal{A} := \{\langle i : C \geq 1\rangle\}$ *for some new individual name* i*.*

The TBox $\mathcal{T}$ of $\pi(\mathrm{T}, C)$ consists of (actual) GCIs only, i.e. it can not be represented by an equivalent finite set of TBox axioms of the form $A \sqsubseteq C$ or $A \equiv C$ s.t. $A \in \mathbf{C}$ (hence $A \neq \top$) and $C \in \mathcal{C}(\Sigma)$. Hence, the reduction uses extensively expressive means that are not available in [21] in which KB satisfiability is shown to be PSPACE-complete if TBoxes are restricted to expressions of the form $A \sqsubseteq C$ or $A \equiv C$ s.t. any concept name A occurs at most once on the left-hand side and the TBox does not contain any cyclic dependencies between concept names.

The intention of the two components $\mathcal{T}_1$ and $\mathcal{T}_2$ that $\mathcal{T}$ consists of is as follows: $\mathcal{T}_2$ ensures that any model of $\mathcal{T}$ (and hence $\pi(\mathrm{T}, C)$) assigns only possibility degrees in $\{0, 1\}$ to any concept (sub-)expression occurring somewhere in T or C. In particular, any such model assigns classical truth values to any concept name $A \in \mathbf{C}$ and any role $R \in \mathbf{R}$ in any situation. Additionally, $\mathcal{T}_1$ ensures that any model of $\mathcal{T}$ assigns the possibility degree 1 to the concepts $\neg D \sqcup D'$ for any individual. Since for possibility degrees in $\{0, 1\}$ the fuzzy semantics of concept constructors coincides with the classical semantics, we know that a fuzzy interpretation that satisfies $\mathcal{T}$ represents in fact a classical (i.e. crisp) model of T. By the same line of argumentation, $\mathcal{A}$ ensures that in a model of $\pi(\mathrm{T}, C)$ (and hence $\mathcal{T}$) the input concept C is satisfiable too.

It is straightforward to see that the reduction $\pi(\mathrm{T}, C)$ can be computed in linear time (wrt. the size of T and C) for each finite set T of GCIs and concept expressions $C \in \mathcal{C}(\Sigma)$. It remains to show that $\pi(\mathrm{T}, C)$ is satisfiable in $\mathbb{ALC}$ iff C is satisfiable wrt. T in $\mathcal{ALC}$.

We start our formal proof by relating particular fuzzy interpretations and crisp interpretations:

[1] One way to show ExpTime-hardness of concept satisfiability wrt. general terminologies in the classical DL $\mathcal{ALC}$ is by encoding the computation of an polynomially space-bounded alternating Turing machine in a respective TBox and use the input concept to specify the start configuration of the computation of that machine. This demonstrates APSPACE-hardness and therefore EXPTIME-hardness, since APSPACE = EXPTIME.

Definition 3.2. *Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a $\mathbb{ALC}$-interpretation and $\mathrm{I} = (\Delta^{\mathrm{I}}, \cdot^{\mathrm{I}})$ be a $\mathcal{ALC}$-interpretation over the same signature Σ. Let T denote a finite set of GCIs over Σ, $D \in \mathcal{C}(\Sigma)$ denote a concept expression, and let $\mathsf{sub}(\mathcal{K})$ denote the set of all concept expressions that occur as subexpressions somewhere in an axiom in a KB $\mathcal{K}$. We say $\mathcal{I}$ and I are* corresponding interpretations wrt. the TBox T and the concept expression D *(denoted by $\mathcal{I} \simeq_{\mathrm{T},D} \mathrm{I}$) iff*

1. $\Delta^{\mathcal{I}} = \Delta^{\mathrm{I}}$ *and*
2. $C^{\mathcal{I}}(o) \in \{0, 1\}$ *for all $o \in \Delta^{\mathcal{I}}$ and concept expressions $C \in \mathsf{sub}(\pi(\mathrm{T}, D))$ and*
3. $A^{\mathcal{I}}(o) = 1$ *iff $o \in A^{\mathrm{I}}$ for all $o \in \Delta^{\mathcal{I}}$ and atomic concept names $A \in \mathbf{C}$ and*
4. $R^{\mathcal{I}}(o, o') > 0$ *iff $\langle o, o' \rangle \in R^{\mathrm{I}}$ for all $o, o' \in \Delta^{\mathcal{I}}$ and role names $R \in \mathbf{R}$ and*

We can now show that corresponding interpretations behave in a some sense analogously under their respective semantics:

Proposition 3.1. *Let T denote a finite set of GCIs over Σ and $D^* \in \mathcal{C}(\Sigma)$ be any concept expression. Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a $\mathbb{ALC}$-interpretation and $\mathrm{I} = (\Delta^{\mathrm{I}}, \cdot^{\mathrm{I}})$ be a $\mathcal{ALC}$-interpretation over Σ such that $\mathcal{I} \simeq_{\mathrm{T},D^*} \mathrm{I}$. Then, $C^{\mathcal{I}}(o) = 1$ iff $o \in C^{\mathrm{I}}$ for all $o \in \Delta^{\mathcal{I}}$ and concept expressions $C \in \mathsf{sub}(\pi(\mathrm{T}, D^*))$.*

Proof. We proof the proposition by induction over the structure of concept expressions $C \in \mathcal{C}(\Sigma)$:

Let $o \in \Delta^{\mathcal{I}}$ denote any individual. For the base case (i.e for concept names $A \in \mathbf{C}$ or $C = \top$) the proposition follows immediately from the fact that $\mathcal{I} \simeq_{\mathrm{T},D^*} \mathrm{I}$ by clause (3) in Def. 3.2. For the induction step we consider the single concept constructors that can be used to form a concept $C \in \mathcal{C}(\Sigma)$ one-by-one. In the following, we can safely assume that $o \in \Delta^{\mathrm{I}}$ by clause (1) in Def. 3.2:

- $C = D \sqcap E$: $1 = C^{\mathcal{I}}(o) = (D \sqcap E)^{\mathcal{I}} = min(D^{\mathcal{I}}(o), E^{\mathcal{I}}(o))$ iff $D^{\mathcal{I}}(o) = 1$ and $E^{\mathcal{I}}(o) = 1$ (since $D^{\mathcal{I}}(o)$ and $D^{\mathcal{I}}(o)$ must both be smaller or equal to 1) iff $o \in D^{\mathrm{I}}$ and $o \in E^{\mathrm{I}}$ (by applying the induction hypothesis to the simpler concept subexpressions $D, E \in \mathsf{sub}(\pi(\mathrm{T}, D^*))$) iff $o \in (D \sqcap E)^{\mathrm{I}} = C^{\mathrm{I}}$.
- $C = D \sqcup E$: works analogously to the case $C = D \sqcap E$.
- $C = \neg D$: $1 = C^{\mathcal{I}}(o) = (\neg D)^{\mathcal{I}} = 1 - D^{\mathcal{I}}(o)$ iff $D^{\mathcal{I}}(o) = 0$ iff $o \notin D^{\mathrm{I}}$ (by applying the induction hypothesis to the simpler concept subexpression $D \in \mathsf{sub}(\pi(\mathrm{T}, D^*))$ and clause (2) in Def. 3.2) iff $o \in (\neg D)^{\mathrm{I}} = C^{\mathrm{I}}$.
- $C = \forall R.D$: Then $D \in \mathsf{sub}(\pi(\mathrm{T}, D^*))$ and D is a simpler concept expression than C. For the direction from left to right assume $1 = C^{\mathcal{I}} = (\forall R.D)^{\mathcal{I}} = inf_{o' \in \Delta^{\mathcal{I}}}(max(1 - R^{\mathcal{I}}(o, o'), D^{\mathcal{I}}(o'))\}$. Hence, for all $o' \in \Delta^{\mathcal{I}}$ it must hold that $1 \leq max(1 - R^{\mathcal{I}}(o, o'), D^{\mathcal{I}}(o'))$ and therefore $1 = max(1 - R^{\mathcal{I}}(o, o'), D^{\mathcal{I}}(o'))$ $(*)$. Assume now that $o \notin C^{\mathrm{I}} = (\forall R.D)^{\mathrm{I}}$. Then, there must exist some $o' \in \Delta^{\mathrm{I}} = \Delta^{\mathcal{I}}$ s.t. $\langle o, o' \rangle \in R^{\mathrm{I}}$ and $o' \notin D^{\mathrm{I}}$ (by the semantics of universal role restriction in $\mathcal{ALC}$). Applying the induction hypothesis to D, we can conclude that there exists a $o' \in \Delta^{\mathcal{I}}$ s.t. $\langle o, o' \rangle \in R^{\mathrm{I}}$ and $D^{\mathcal{I}}(o') \neq 1$. From clauses (2) and (4) we can infer that $R^{\mathcal{I}}(o, o') > 0$ and $D^{\mathcal{I}}(o') = 0$. Hence, for this specific $o' \in \Delta^{\mathcal{I}}$, we know that $max(1 - R^{\mathcal{I}}(o, o'), D^{\mathcal{I}}(o')) = 1 - R^{\mathcal{I}}(o, o') < 1$ which contradicts $(*)$. Therefore, our assumption must be wrong and $o \notin C^{\mathrm{I}}$ must hold.
  For the direction from right to left, let $o \in C^{\mathrm{I}}$ hold, i.e. for all $o' \in \Delta^{\mathrm{I}}$ it holds that $\langle o, o' \rangle \in R^{\mathrm{I}}$ implies $o' \in D^{\mathrm{I}}$. Assume that $1 \neq C^{\mathcal{I}}(o)$, hence $C^{\mathcal{I}}(o) = inf_{o' \in \Delta^{\mathcal{I}}}(max(1 - R^{\mathcal{I}}(o, o'), D^{\mathcal{I}}(o'))\} < 1$. Hence, there must exist a $o' \in \Delta^{\mathcal{I}} = \Delta^{\mathrm{I}}$ s.t. $max(1 - R^{\mathcal{I}}(o, o'), D^{\mathcal{I}}(o'))) < 1$. Hence, for this $o' \in \Delta^{\mathcal{I}}$ both $1 - R^{\mathcal{I}}(o, o') < 1$ and $D^{\mathcal{I}}(o') < 1$ must hold. Because of clause (2), this means that $R^{\mathcal{I}}(o, o') > 0$ and $D^{\mathcal{I}}(o') = 0$ must hold for $o' \in \Delta^{\mathcal{I}}$. Applying the induction hypothesis to D and considering the clauses (1) and (4) in Def. 3.2 we can conclude that there exists an $o' \in \Delta^{\mathrm{I}}$ s.t.

$o' \notin D^{\mathrm{I}}$ and $\langle o, o' \rangle \in R^{\mathrm{I}}$, which contradicts that $o \in (\forall R.D)^{\mathrm{I}} = C^{\mathrm{I}}$. Hence, again our assumption must be wrong and therefore $C^{\mathcal{I}}(o) = 1$.

□

Using this technical proposition, we can now proceed and show that our mapping $\pi(\mathrm{T}, C)$ in fact reduces the problem to determine concept satisfiability of C wrt. a general TBox T in $\mathcal{ALC}$ to KB satisfiability of $\pi(\mathrm{T}, C)$ in $\mathbb{ALC}$.

Proposition 3.2. *Let* T *denote a finite set of GCIs and* $C \in \mathcal{C}(\Sigma)$ *be any concept expression. Let,* $\pi(\mathrm{T}, C) = (\mathcal{T}, \mathcal{A})$ *be* $\mathbb{ALC}$ *KB which is the represents the reduction of the terminology* T *and the concept expression* C *as defined in Def. 3.1. Then,* $\pi(\mathrm{T}, C)$ *is satisfiable in* $\mathbb{ALC}$ *if* C *is satisfiable wrt.* T *in* $\mathcal{ALC}$.

Proof. Let C be satisfiable wrt. T in $\mathcal{ALC}$ and I denote a (crisp) interpretation such that $\mathrm{I} \models \mathrm{T}$ and $o^* \in C^{\mathrm{I}}$ for some $o^* \in \Delta^{\mathrm{I}}$. We define a (corresponding) fuzzy interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ as follows:

$$\Delta^{\mathcal{I}} := \Delta^{\mathrm{I}} \quad R^{\mathcal{I}}(o, o') := \begin{cases} 1 \text{ if } \langle o, o' \rangle \in R^{\mathrm{I}} \\ 0 \text{ if } \langle o, o' \rangle \notin R^{\mathrm{I}} \end{cases} \quad A^{\mathcal{I}}(o) := \begin{cases} 1 \text{ if } o \in A^{\mathrm{I}} \\ 0 \text{ if } o \notin A^{\mathrm{I}} \end{cases} \quad i^{\mathcal{I}} := o^*$$

for any $o, o' \in \Delta^{\mathcal{I}}$, $R \in \mathbf{R}$ and $A \in \mathbf{C}$.

Our definition is well-defined for all atomic symbols (i.e. concept names, role names and individual names) that occur in $\pi(\mathrm{T}, C)$. It is easy to see that $\mathcal{I} \simeq_{\mathrm{T},C} \mathrm{I}$: clauses (1), (3), and (4) in Def. 3.2 are satisfied immediately by our definition of $\mathcal{I}$. Clause (2) is satisfied for concept names $A \in \mathbf{C}$ as well. Since $\mathcal{I}$ only assigns possibility degrees $R^{\mathcal{I}}(o, o')$ and $A^{\mathcal{I}}(o)$ in $\{0, 1\}$, it follows by a trivial induction argument that any possibility degree $C^{\mathcal{I}}(o)$ assigned to any $o \in \mathcal{I}$ must be in $\{0, 1\}$ too.

We now show that $\mathcal{I} \models \mathcal{T}$: let $\alpha = \top \sqsubseteq \neg D \sqcup D'$ be any TBox axiom $\alpha \in \mathcal{T}_1$ (and therefore $D \sqsubseteq D' \in \mathrm{T}$). Then, $\mathcal{I} \models \alpha$ iff $(\neg D \sqcup D')^{\mathcal{I}}(o) = 1$ for all $o \in \Delta^{\mathcal{I}}$. Because $\mathcal{I} \simeq_{\mathrm{T},C} \mathrm{I}$, we can use Proposition 3.1 and conclude that $o \in (\neg D \sqcup D')^{\mathrm{I}}$ for all $o \in \Delta^{\mathrm{I}}$ iff $o \in D^{\mathrm{I}}$ implies $o \in (D')^{\mathrm{I}}$ for all $o \in \Delta^{\mathrm{I}}$ iff $D^{\mathrm{I}} \subseteq (D')^{\mathrm{I}}$. The latter holds, since I is a model of T and therefore $\mathrm{I} \models D \sqsubseteq D'$.

Let $\alpha = \top \sqsubseteq \neg D \sqcup D$ be any TBox axiom $\alpha \in \mathcal{T}_2$ (and therefore $D \in \mathsf{sub}(\mathrm{T} \cup \{C\})$). Then, $\mathcal{I} \models \alpha$ iff $(\neg D \sqcup D)^{\mathcal{I}}(o) = 1$ for all $o \in \Delta^{\mathcal{I}}$. This is the case iff $max(1 - D^{\mathcal{I}}(o), D^{\mathcal{I}}(o)) = 1$ for all $o \in \Delta^{\mathcal{I}}$ iff $D^{\mathcal{I}}(o) \in \{0, 1\}$ for all $o \in \Delta^{\mathcal{I}}$. The latter is satisfied by clause (2) of Def. 3.2, since $\mathcal{I} \simeq_{\mathrm{T},C} \mathrm{I}$ and $\mathsf{sub}(\mathrm{T} \cup \{C\}) \subseteq \mathsf{sub}(\pi(\mathrm{T}, C))$ (by our construction of $\pi(\mathrm{T}, C)$ in Def. 3.1). Since $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$ we know that $\mathcal{I} \models \mathcal{T}$.

Further, we can show that $\mathcal{I} \models \mathcal{A}$: $\mathcal{A} = \{\langle i : C \geq 1 \rangle\}$, hence we only need to show that $\mathcal{I} \models \langle i : C \geq 1 \rangle$. This is the case iff $C^{\mathcal{I}}(i^{\mathcal{I}}) = 1$. Because $\mathcal{I} \simeq_{\mathrm{T},C} \mathrm{I}$ and $i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, we can use Proposition 3.1 and conclude that $C^{\mathcal{I}}(i^{\mathcal{I}}) = 1$ iff $i^{\mathrm{I}} \in C^{\mathrm{I}}$ which is satisfied since $i^{\mathcal{I}}$ has been chosen as some $o^* \in \Delta^{\mathrm{I}} = \Delta^{\mathcal{I}}$ s.t. $o^* \in C^{\mathrm{I}}$.

In summary, we have shown that $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$ which proofs $\mathcal{I} \models \pi(\mathrm{T}, C)$. Hence, $\mathcal{I}$ is a model of $\pi(\mathrm{T}, C)$ and $\pi(\mathrm{T}, C)$ is satisfiable in $\mathbb{ALC}$. □

Proposition 3.3. *Let* T *denote a finite set of GCIs and* $C \in \mathcal{C}(\Sigma)$ *be any concept expression. Let,* $\pi(\mathrm{T}, C)$ *be* $\mathbb{ALC}$ *KB which represents the reduction of the terminology* T *and the concept expression* C *as defined in Def. 3.1. Then,* C *is satisfiable wrt.* T *in* $\mathcal{ALC}$ *if* $\pi(\mathrm{T}, C)$ *is satisfiable in* $\mathbb{ALC}$.

Proof. Let $\pi(\mathrm{T}, C)$ be satisfiable in $\mathbb{ALC}$ and $\mathcal{I}$ denote a (fuzzy) interpretation such that $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$. We define a (corresponding) crisp interpretation $\mathrm{I} = (\Delta^{\mathrm{I}}, \cdot^{\mathrm{I}})$ as follows:

$$\Delta^{\mathrm{I}} := \Delta^{\mathcal{I}} \quad R^{\mathrm{I}} := \{\langle o, o' \rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(o, o') > 0\} \quad A^{\mathrm{I}} := \{o \mid A^{\mathcal{I}}(o) = 1\}$$

for any $R \in \mathbf{R}$ and $A \in \mathbf{C}$.

Our definition is well-defined for all atomic symbols (i.e. concept names, role names and individual names) that occur in $\pi(\mathrm{T}, C)$.

We first show that $\mathcal{I} \simeq_{\mathrm{T},c} \mathrm{I}$: clauses (1), (3), and (4) in Def. 3.2 are satisfied immediately by our definition of I. It remains to show that clause (2) is satisfied for concept expressions $E \in \mathsf{sub}(\pi(\mathrm{T}, C))$ as well: by Def. 3.1, $\mathsf{sub}(\pi(\mathrm{T}, C)) = \{\top\} \cup \{\neg D \sqcup D \mid D \in \mathsf{sub}(\mathrm{T} \cup \{C\})\} \cup \{\neg D \mid D \in \mathsf{sub}(\mathrm{T} \cup \{C\})\} \cup \{\neg D \mid D \sqsubseteq D' \in \mathrm{T}\} \cup \mathsf{sub}(\mathrm{T} \cup \{C\})$.

For $E = \top$, the clause (2) is trivially satisfied. For $E = \neg D \sqcup D$ such that $D \in \mathsf{sub}(\mathrm{T} \cup \{C\})$ and for $E \in \mathsf{sub}(\mathrm{T} \cup \{C\})$ clause (2) is satisfied since $\mathcal{I} \models \mathcal{T}$, hence $\mathcal{I} \models \mathcal{T}_2$ and we therefore know that $1 \leq E^{\mathcal{I}}(o) = (\neg D \sqcup D)^{\mathcal{I}}(o) = max(1 - D^{\mathcal{I}}(o), D^{\mathcal{I}}(o)) \leq 1$ for each $o \in \Delta^{\mathcal{I}}$ and each $D \in \mathsf{sub}(\mathrm{T} \cup \{C\})$. In other words, for each $o \in \Delta^{\mathcal{I}}$ and each $D \in \mathsf{sub}(\mathrm{T} \cup \{C\})$ we know that $E^{\mathcal{I}}(o) = max(1 - D^{\mathcal{I}}(o), D^{\mathcal{I}}(o)) = 1$, which holds iff $D^{\mathcal{I}}(o) \in \{0, 1\}$. For $E = \neg D$ for some $D \in \mathsf{sub}(\mathrm{T} \cup \{C\})$, we can derive that $E^{\mathcal{I}}(o) = (\neg D)^{\mathcal{I}}(o) = 1 - D^{\mathcal{I}}(o) \in \{0, 1\}$, since we already know that $D^{\mathcal{I}}(o) \in \{0, 1\}$ must hold for for any $o \in \Delta^{\mathcal{I}}$ and any such a concept expression D. Hence clause (2) is satisfied in this case too. The last case to consider is that $E = \neg D$ for some $D \sqsubseteq D' \in \mathrm{T}$. But then, D is a concept subexpression of T, i.e. $D \in \mathsf{sub}(\mathrm{T} \cup \{C\})$ for which we already know that $D^{\mathcal{I}}(o) \in \{0, 1\}$ must hold for for any $o \in \Delta^{\mathcal{I}}$. Consequently, $E^{\mathcal{I}}(o) = (\neg D)^{\mathcal{I}}(o) = 1 - D^{\mathcal{I}}(o) \in \{0, 1\}$ for any $o \in \Delta^{\mathcal{I}}$ in this case too and clause (2) is satisfied. Finally, we have established $\mathcal{I} \simeq_{\mathrm{T},c} \mathrm{I}$.

Since $\mathcal{I} \models \mathcal{T}$, we now that for any $D \sqsubseteq D' \in \mathrm{T}$ and any $o \in \Delta^{\mathcal{I}}$ it must hold that $(\neg D \sqcup D')^{\mathcal{I}}(o) = 1$. Because $\mathcal{I} \simeq_{\mathrm{T},C} \mathrm{I}$ and $i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, we can use Proposition 3.1 and conclude that $(\neg D \sqcup D')^{\mathcal{I}}(o) = 1$ iff $o \in (\neg D \sqcup D')^{\mathrm{I}}$. Hence, for any $o \in \Delta^{\mathcal{I}}$ it must hold that $o \in D^{\mathrm{I}}$ implies $o \in (D')^{\mathrm{I}}$, which is equivalent to $D^{\mathrm{I}} \subseteq (D')^{\mathrm{I}}$. Consequently, $\mathrm{I} \models D \sqsubseteq D'$ for any $D \sqsubseteq D' \in \mathrm{T}$, i.e. $\mathrm{I} \models \mathcal{T}$.

Since $\mathcal{I} \models \mathcal{A}$, we know that $\mathcal{I} \models \langle i : C \geq 1 \rangle$. This is the case iff $C^{\mathcal{I}}(i^{\mathcal{I}}) = 1$. Because $\mathcal{I} \simeq_{\mathrm{T},C} \mathrm{I}$ and $i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, we can use Proposition 3.1 and conclude that $C^{\mathcal{I}}(i^{\mathcal{I}}) = 1$ iff $i^{\mathrm{I}} \in C^{\mathrm{I}}$. By clause (1) of Def. 3.2, we conclude that there exists and $o \in \Delta^{\mathrm{I}}$ s.t. $o \in C^{\mathrm{I}}$, i.e. choose $o = i^{\mathcal{I}} \in \Delta^{\mathcal{I}} = \Delta^{\mathrm{I}}$.

In summary, we have shown that I is a model of T for which C is interpreted non-empty. Hence, C is satisfiable wrt. T in $\mathcal{ALC}$. □

Proposition 3.2 and Proposition 3.3 together proof that the transformation $\pi(\mathrm{T}, C)$ described in Def.3.1 is a reduction of concept satisfiability wrt. general TBoxes in $\mathcal{ALC}$ to the problem of satisfiability of unrestricted KBs in $\mathbb{ALC}$. Since we know that the transformation $\pi(\mathrm{T}, C)$ described in Def. 3.1 can be computed in polynomial time, we have a polynomial time reduction of an EXPTIME-hard problem to KB satisfiability in $\mathbb{ALC}$. For this reason KB satisfiability in $\mathbb{ALC}$ (when GCIs are allowed) must be an EXPTIME-hard computational problem too.

Theorem 3.4 (EXPTIME-Hardness of KB Satisfiability)**.** *The problem of deciding the satisfiability of KBs in* $\mathbb{ALC}$ *is* EXPTIME*-hard if GCIs are allowed.*

ExpTime-Membership. KB satisfiability is in EXPTIME since [22] shows that checking KB satisfiability in $\mathbb{ALC}$ can be reduced (in polynomial time) to checking KB satisfiability in $\mathcal{ALC}$ which is known to be in EXPTIME, since KB satisfiability is in EXPTIME even for an extension of $\mathcal{ALC}$, i.e. the more expressive DL $\mathcal{SHIQ}$ [25, Corollary 6.30].

Therefore we can characterize the worst-case complexity of a fundamental reasoning problem in $\mathbb{ALC}$ as follows:

Theorem 3.5 (EXPTIME-Completeness of KB Satisfiability)**.** *The problem of deciding the satisfiability of KBs in $\mathbb{ALC}$ is* EXPTIME*-complete if GCIs are allowed.*

Remark 3.1 (EXPTIME-Completeness of KB Satisfiability in restricted cases)**.** *Our proof indeed gives us a stronger result: the reduction used in the proof above (see Def.3.1) shows that* EXPTIME*-completeness of KB satisfiability in $\mathbb{ALC}$ holds even for KBs which consist only of GCIs of the form $\top \sqsubseteq C$ and of a single ABox axiom of the form $\langle i : C \geq 1\rangle$ for $C \in \mathcal{C}(\Sigma)$ – in other words, for KBs where the only possibility degree that is explicitly mentioned in the KB is* 1*, only a single concept membership axiom is included in the ABox and GCIs use exclusively the atomic concept expression $\top$ on the left-hand side.*

3.2 Terminological Reasoning

Let us now consider reasoning tasks which are mainly concerned with the terminological level of a KB, i.e. reasoning tasks which query (in some way) the conceptual schema represented by the TBox.

We introduced three such reasoning tasks in Section 2 that reflect the most basic set-theoretic relations between (fuzzy) sets: given a KB $\mathcal{K}$, check for concept subsumption $C \sqsubseteq D$ wrt. $\mathcal{K}$, concept equivalence $C \equiv D$ wrt. $\mathcal{K}$, concept disjointness wrt. $\mathcal{K}$, and concept satisfiability wrt. $\mathcal{K}$. We will discuss the worst-case complexity for each of these terminological reasoning tasks one-by-one.

Entailment of Concept Subsumption. KB satisfiability can be reduced in polynomial time (via a many-one-reduction) to non-entailment of a concept subsumption: KB $\mathcal{K}$ is satisfiable iff $\mathcal{K} \not\models \top \sqsubseteq \bot$. Therefore, checking non-entailment of concept subsumption is an EXPTIME-hard problem. Consequently, checking for entailment of concept subsumption is a CO-EXPTIME-hard problem, and since CO-EXPTIME = EXPTIME, we can conclude that it is also EXPTIME-hard. EXPTIME-membership for entailment of concept subsumption can be shown by a polynomial time turing-reduction to KB satisfiability:

$\mathcal{K} \models C \sqsubseteq D$ iff it does not holds that there exists a model $\mathcal{I}$ of $\mathcal{K}$ such that there exist $o \in \Delta^{\mathcal{I}}$ and $d, d' \in [0, 1]$ such that $C^{\mathcal{I}}(o) = d$, $D^{\mathcal{I}}(o) = d'$ and $d > d'$.

It has been shown in [21, 22] that if $\mathcal{K}$ is satisfiable, then there is as well a model of $\mathcal{K}$ which assigns possibility degrees in $\mathsf{PossDeg}(\mathcal{K})$ only. Hence, we have $\mathcal{K} \models C \sqsubseteq D$ iff it does not holds that there exists a model $\mathcal{I}$ of $\mathcal{K}$ such that there exist $o \in \Delta^{\mathcal{I}}$ and $d, d' \in \mathsf{PossDeg}(\mathcal{K})$ such that $C^{\mathcal{I}}(o) = d$, $D^{\mathcal{I}}(o) = d'$ and $d > d'$.

In other words, given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and any $d, d' \in \mathsf{PossDeg}(\mathcal{K})$ such that $d > d'$, the modified KB $\mathcal{K}_{d,d'} = (\mathcal{T}, \mathcal{A}_{d,d'})$ with $\mathcal{A}_{d,d'} = \mathcal{A} \cup \{\langle i : D = d\rangle, \langle i : C = d'\rangle\}$ for some new individual name i not occurring in $\mathcal{K}$ must be unsatisfiable. For any $d, d' \in \mathsf{PossDeg}(\mathcal{K})$ the modified KB $\mathcal{K}_{d,d'}$ can be constructed in polynomial time. Further, $\mathsf{PossDeg}(\mathcal{K})$ is a finite set with $|\mathsf{PossDeg}(\mathcal{K})| \in O(|\mathcal{K}|)$. Hence, there are at most $O(|\mathcal{K}|^2)$ many pairs d, d' that need to be considered. For each pair $d, d' \in \mathsf{PossDeg}(\mathcal{K})$ such that $d > d'$ we construct the modified modified KB $\mathcal{K}_{d,d'}$ and check for unsatisfiability. If any of the constructed KBs $\mathcal{K}_{d,d'}$ is satisfiable, we know that $\mathcal{K} \not\models C \sqsubseteq D$. If the polynomial number of modified KBs all are unsatisfiable, we know that $\mathcal{K} \models C \sqsubseteq D$. Overall, we have shown EXPTIME-completeness for entailment of concept subsumption wrt. general KBs.

Entailment of Concept Equivalence. KB satisfiability can be reduced in polynomial time (via a many-one-reduction) to non-entailment of a concept equivalence: KB $\mathcal{K}$ is satisfiable iff $\mathcal{K} \not\models \top \equiv \bot$. Therefore,

checking non-entailment of concept equivalences is an EXPTIME-hard problem. Consequently, checking for entailment of concept equivalences is a CO-EXPTIME-hard problem, and since CO-EXPTIME = EXPTIME, we can conclude that it is also EXPTIME-hard. Further, entailment of concept equivalence can be reduced in polynomial time to a pair of concept subsumption problems: $\mathcal{K} \models C \equiv D$ iff $\mathcal{K} \models C \sqsubseteq D$ and $\mathcal{K} \models D \sqsubseteq C$. Hence, EXPTIME-membership for concept equivalence wrt. general KBs follows from the EXPTIME-membership for concept subsumption wrt. general KBs in $\mathbb{ALC}$. Overall, we established have EXPTIME-completeness for entailment of concept equivalence wrt. general KBs.

Entailment of Concept Disjointness. Concept disjointness can be reduced in polynomial time (via a many-one-reduction) to a concept subsumption problem: two concepts C and D are disjoint wrt. $\mathcal{K}$ iff $\mathcal{K} \models C \sqcap D \sqsubseteq \bot$ Hence, EXPTIME-hardness for concept disjointness wrt. general KBs follows from the EXPTIME-hardness for concept subsumption wrt. general KBs in $\mathbb{ALC}$. Further, entailment of concept disjointness can be reduced in polynomial time to KB satisfiability: two concepts C and D are disjoint wrt. the KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ iff it is not the case that the modified KB $\mathcal{K}_\epsilon = (\mathcal{T}, \mathcal{A}_\epsilon)$ with $\mathcal{A}_\epsilon = \mathcal{A} \cup \{\langle i : C \sqcap D \geq \epsilon \rangle\}$ for some new individual name i not occurring in $\mathcal{K}$ is unsatisfiable for some $0 < \epsilon \leq 1$. Again, we can restrict our attention to the finitely many possibility degrees $d \in \mathsf{PossDeg}(\mathcal{K})$. Therefore, we can determine the disjointness of two concepts by a single KB satisfiability test: choose some positive $\epsilon \in (0, 1]$ s.t. $\epsilon \leq min(\mathsf{PossDeg}(\mathcal{K}))$, construct $\mathcal{K}_\epsilon$ (in polynomial time) and check if $\mathcal{K}_\epsilon$ is satisfiable. If so, we know that C and D are not disjoint wrt. $\mathcal{K}$. If $\mathcal{K}_\epsilon$ is unsatisfiable, we know that C and D are disjoint wrt. $\mathcal{K}$. Overall, we have shown EXPTIME-completeness for entailment of concept disjointness wrt. general KBs.

Concept Satisfiability. KB satisfiability can be reduced in polynomial time (via a many-one-reduction) to a concept satisfiability problem (wrt. a general KB): $\mathcal{K}$ is satisfiable iff $\top$ is satisfiable wrt. $\mathcal{K}$. Hence, EXPTIME-hardness for concept satisfiability wrt. general KBs follows from the EXPTIME-hardness for KB satisfiability in $\mathbb{ALC}$ if GCIs are allowed. Further, concept satisfiability wrt. general KBs can be reduced in polynomial time to non-entailment of concept subsumption wrt. general KBs as follows: a concept C is satisfiable wrt. a KB $\mathcal{K}$ iff $\mathcal{K} \not\models C \sqsubseteq \bot$. Since entailment of concept subsumption wrt. general KBs is in EXPTIME, it follows that concept satisfiability wrt. general KBs is in EXPTIME too. Overall, we have shown EXPTIME-completeness for concept satisfiability wrt. general KBs.

We can summarize the complexity results for terminological reasoning as follows:

Theorem 3.6 (Complexity of Terminological Reasoning)**.** *Any of the following terminological reasoning tasks in $\mathbb{ALC}$ is* EXPTIME*-hard if GCIs are allowed: checking for concept equivalence wrt. a KB, checking for concept subsumption wrt. a KB, checking for concept disjointness wrt. a KB, and checking for concept satisfiability wrt. a KB.*

3.3 Extensional Reasoning

Let us now consider reasoning tasks which are mainly concerned with the assertional level of a KB, i.e. reasoning tasks which query (in some way) structure represented in the ABox, i.e. the instances and their interrelation. In contrast to purely structure-based data models (such as the Relational Data Model underlying RDBMS), the implicit knowledge on the relation of concepts captured by the TBox is hereby taken into account.

We introduced the following reasoning tasks in Section 2:

Entailment of Fuzzy Concept Membership Assertions of the form $\langle i : C \geq n\rangle$. The problem of checking the satisfiability of a KB $\mathcal{K}$ in $\mathbb{ALC}$ can be reduced via a polynomial-time many-one reduction to a fuzzy concept membership entailment problem of the form $\langle i : C \geq n\rangle$: $\mathcal{K}$ is satisfiable iff $\mathcal{K} \not\models \langle i : \bot \geq 1\rangle$ for a new individual i not appearing in $\mathcal{K}$. Consequently, checking for entailment of fuzzy concept membership of the form $\langle i : C \geq n\rangle$ is a CO-EXPTIME-hard problem, and since CO-EXPTIME = EXPTIME, we can conclude that it is also EXPTIME-hard. EXPTIME-membership for entailment of a fuzzy concept membership of the form $\langle i : C \geq n\rangle$ can be shown by a polynomial-time turing-reduction to KB satisfiability: consider a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and let $\mathcal{K}_\epsilon = (\mathcal{T}, \mathcal{A}_\epsilon)$ be a modified KB such that $\mathcal{A}_\epsilon = \mathcal{A} \cup \{\langle i : C \leq n - \epsilon\rangle\}$. Then, $\mathcal{K} \models \langle i : C \geq n\rangle$ iff $\mathcal{K}_\epsilon$ is not satisfiable for some (suitably small) positive $\epsilon > 0$. In order to find a sufficiently small $\epsilon > 0$ for a given KB $\mathcal{K}$, we exploit again the property that a KB $\mathcal{K}$ is satisfiable iff there exists a model of $\mathcal{K}$ which assigns possibility degrees in the finite set $\mathsf{PossDeg}(\mathcal{K})$ only. We simply select any positive rational number $\epsilon > 0$ such that $\epsilon \leq min\{d - d' \mid d, d' \in \mathsf{PossDeg}(\mathcal{K}), d > d'\}$. Clearly, such a suitable ϵ can be found in $O(|\mathcal{K}|^2)$ computation steps. It is easy to verify that for any ϵ chosen this way it holds that $\mathcal{K} \models \langle i : C \geq n\rangle$ iff $\mathcal{K}_\epsilon$ is not satisfiable. Since the determining the satisfiability of a KB is in EXPTIME and $\mathcal{K}_\epsilon$ can be constructed in polynomial time wrt. the size of input KB $\mathcal{K}$, we can decide the entailment of a fuzzy concept membership of the form $\langle i : C \geq n\rangle$ in exponential time wrt. the size of input KB $\mathcal{K}$. Overall, we have shown EXPTIME-completeness for a fuzzy concept membership entailment problems of the form $\langle i : C \geq n\rangle$.

Entailment of Fuzzy Concept Membership Assertions of the form $\langle i : C \leq n\rangle$. Entailment of fuzzy ABox assertions of the form $\langle i : C \geq n\rangle$ can be reduced via a polynomial-time many-one reduction to the entailment of fuzzy ABox assertions of the form $\langle i : C \leq n\rangle$ because $\langle i : C \geq n\rangle$ and $\langle i : \neg C \leq 1 - n\rangle$ are equivalent fuzzy ABox assertions, i.e. for any interpretation $\mathcal{I}$ it holds that $\mathcal{I} \models \langle i : C \geq n\rangle$ iff $\mathcal{I} \models \langle i : \neg C \leq 1 - n\rangle$. Therefore, $\mathcal{K} \models \langle i : C \geq n\rangle$ iff $\mathcal{K} \models \langle i : \neg C \leq 1 - n\rangle$ which shows EXPTIME-HARDNESS of checking for the entailment of fuzzy ABox assertions of the form $\langle i : C \leq n\rangle$. EXPTIME-membership for entailment of a fuzzy concept membership of the form $\langle i : C \leq n\rangle$ can be shown in the same way by a polynomial-time many-one reduction to entailment of a fuzzy concept membership of the form $\langle i : C \geq n\rangle$: $\mathcal{K} \models \langle i : C \leq n\rangle$ iff $\mathcal{K} \models \langle i : \neg C \geq 1 - n\rangle$. Overall, we have shown EXPTIME-completeness for a fuzzy concept membership entailment problems of the form $\langle i : C \leq n\rangle$.

Entailment of Fuzzy Concept Membership Assertions of the form $\langle i : C = n\rangle$. The problem of checking the satisfiability of a KB $\mathcal{K}$ in $\mathbb{ALC}$ can be reduced via a polynomial-time many-one reduction to a fuzzy concept membership entailment problem of the form $\langle i : C = n\rangle$: $\mathcal{K}$ is satisfiable iff $\mathcal{K} \not\models \langle i : \bot = 1\rangle$ for a new individual i not appearing in $\mathcal{K}$. Consequently, checking for entailment of fuzzy concept membership of the form $\langle i : C = n\rangle$ is a CO-EXPTIME-hard problem, and since CO-EXPTIME = EXPTIME, we can conclude that it is also EXPTIME-hard. EXPTIME-membership for entailment of a fuzzy concept membership of the form $\langle i : C = n\rangle$ can be shown by a polynomial-time turing-reduction to the entailment of a fuzzy concept membership of the form $\langle i : C \geq n\rangle$: $\mathcal{K} \models \langle i : C = n\rangle$ iff $\mathcal{K} \models \langle i : C \geq n\rangle$ and $\mathcal{K} \models \langle i : C \leq n\rangle$ iff $\mathcal{K} \models \langle i : C \geq n\rangle$ and $\mathcal{K} \models \langle i : \neg C \geq 1 - n\rangle$. Overall, we have shown EXPTIME-completeness for a fuzzy concept membership entailment problems of the form $\langle i : C = n\rangle$.

REASONING TASK	WORST-CASE COMPLEXITY
KB satisfiability	EXPTIME-complete
Terminological Reasoning	
$\mathcal{K} \models C \equiv D$	EXPTIME-complete
$\mathcal{K} \models C \sqsubseteq D$	EXPTIME-complete
Concept Disjointness wrt. $\mathcal{K}$	EXPTIME-complete
Concept Satisfiability wrt. $\mathcal{K}$	EXPTIME-complete
Extensional Reasoning	
$\mathcal{K} \models \langle o : C \geq n \rangle$	EXPTIME-complete
$\mathcal{K} \models \langle o : C \leq n \rangle$	EXPTIME-complete
$\mathcal{K} \models \langle o : C = n \rangle$	EXPTIME-complete
$\mathcal{K} \models \langle R(o, o') \geq n \rangle$	EXPTIME-hard

Figure 3: Worst-case Complexity of Reasoning with general KBs in $\mathbb{ALC}$.

Entailment of Fuzzy Relation Assertions. KB satisfiability in $\mathbb{ALC}$ can be reduced via a polynomial-time many-one reduction to the non-entailment of fuzzy relation assertions of the form $\langle R(o, o) \geq n \rangle$ as follows: let R be some role name not occurring in $\mathcal{K}$ and let i, i' be individual names not occurring in $\mathcal{K}$. Consider some $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and the respective modified $\mathcal{K}' = (\mathcal{T}', \mathcal{A})$ with $\mathcal{T}' = \mathcal{T} \cup \{\exists R.\top \sqsubseteq \bot\}$. Then, it holds that $\mathcal{K}$ is satisfiable iff $\mathcal{K}' \not\models \langle R(i, i') \geq 1 \rangle$. Therefore, checking non-entailment of fuzzy relation assertions $\langle R(o, o') \geq n \rangle$ is an EXPTIME-hard problem. Consequently, checking for entailment of fuzzy relation assertions $\langle R(o, o') \geq n \rangle$ is a CO-EXPTIME-hard problem, and since CO-EXPTIME = EXPTIME, we can conclude that it is also EXPTIME-hard.

We can summarize the complexity results for extensional reasoning as follows:

Theorem 3.7 (Complexity of Extensional Reasoning)**.** *Any of the following extensional reasoning tasks in* $\mathbb{ALC}$ *is* EXPTIME*-complete if GCIs are allowed: entailment of fuzzy concept membership assertions of any of the forms* $\langle i : C \bowtie n \rangle$ *with* $\bowtie \in \{\leq, \geq, =\}$*. Checking the entailment of fuzzy relation assertions of the form* $\langle R(o, o') \geq n \rangle$ *is* EXPTIME*-hard if GCIs are allowed.*

Figure 3 summarizes all results presented above.

4 Reasoning with Terminologies in Fuzzy Tableau-based Methods

Tableau-methods can be used to determine the satisfiability of KBs. In order to detect the satisfiability status of a given KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, they construct tree-structured labeled graphs (i.e. completion forests). The nodes in a completion graph represent individuals in an interpretation $\mathcal{I}$ for $\mathcal{K}$ and the edges capture the interrelation of individuals via roles in $\mathcal{K}$. Node labels capture (bounds on) the degree to which the individual represented by the node is a member of a concept, edge labels specify (bounds on) the degree of interrelation of the connected individuals wrt. roles. The completion graph is initialized to represent the ABox $\mathcal{A}$. Then, it is stepwise extended by analyzing concept labels of nodes and interrelation. An expansion can created new labels for nodes and edges and insert new nodes and edges into the graph. The extension process follows a set of so-called completion rules which are exhaustively applied. During the

completion process non-deterministic choices can appear: there might be more than one way to extent the graph further, but we can not tell which of these extensions will eventually lead to a successful construction of a model for $\mathcal{K}$ (if there is one). In order to stay complete, all possible choices (for any node and any node label) need to be considered as long as no model has been constructed yet. The latter can be determined by checking labels for elementary contradictions (in regard of the specified bounds on membership degrees). If no completion rules are applicable anymore and there are no labels to nodes and edges which contain an elementary contradiction (i.e. we have a fully-expanded and clash-free completion graph), then we found in fact a (witness for a) model for $\mathcal{K}$. If no such graph can be constructed at all, we know that $\mathcal{K}$ is unsatisfiable.

Example. Consider the ABox $\mathcal{A} = \{\langle i : \forall R.\neg B \sqcup \exists R.(B \sqcap C) \leq 0.4\rangle, \langle i : \forall R.C \geq 0.7\rangle\}$. The completion process starts with the completion graph

$$L_0 = \{\langle \forall R.\neg B \sqcup \exists R.(B \sqcap C) \leq 0.4\rangle, \langle \forall R.C \geq 0.7\rangle\}$$
$$\bullet_{\text{i}}$$

Since i can be a member of $\forall R.\neg B \sqcup \exists R.(B \sqcap C)$ only up to degree 0.4, we can derive upper bounds on the possibility degree for i wrt. $\forall R.\neg B$ and $\exists R.(B \sqcap C)$ directly from the semantics of the concept constructor $\sqcup$ and update the label by two new constraints

$$L_1 = L_0 \cup \{\langle \forall R.\neg B \leq 0.4\rangle, \langle \exists R.(B \sqcap C) \leq 0.4\rangle\}$$
$$\bullet_{\text{i}}$$

In order to satisfy the constraint $\langle \forall R.\neg B \leq 0.4\rangle$, we need to introduce a new R-successor node j, a constraint on the degree of R-interrelation between i and j and a constraint on the degree of membership of j in $\neg B$. Both constraints follow immediately from the semantics of the universal role restriction in $\mathbb{ALC}$:

$$\overset{L_1}{\bullet_{\text{i}}} \xrightarrow{\{\langle R \geq 0.6\rangle\}} \overset{L'_0 = \{\langle \neg B \leq 0.4\rangle\}}{\bullet_{\text{j}}}$$

In order to satisfy the constraint $\langle \forall R.C \geq 0.7\rangle \in L_1$ we need to add the constraint $\langle C \geq 0.7\rangle$ to L'_0 (since the edge constraint $\langle R \geq 0.6\rangle$ and the upper bound $\langle R \leq 0.3\rangle$ (derivable from the semantics of the universal role restriction in $\mathbb{ALC}$) are inconsistent with each other). Further, in $\mathbb{ALC}$ $\langle \neg B \leq 0.4\rangle \in L'_0$ can be rewritten as $\langle B \geq 0.6\rangle$

$$\overset{L_1}{\bullet_{\text{i}}} \xrightarrow{\{\langle R \geq 0.6\rangle\}} \overset{L'_1 = \{\langle B \geq 0.6\rangle, \langle C \geq 0.7\rangle\}}{\bullet_{\text{j}}}$$

Finally, the only constraint in the completion graph that we did not consider yet is $\langle \exists R.(B \sqcap C) \leq 0.4\rangle \in I_0$. In fact, the concept expression is not guaranteed to be satisfied given the constraint system and interconnection represented by the current completion graph. To ensure the satisfaction of this concept expression, we need to add $\langle B \sqcap C \leq 0.4\rangle$ (since the edge constraint $\langle R \geq 0.6\rangle$ and the upper bound $\langle R \leq 0.4\rangle$ (derivable from the semantics of the universal role restriction in $\mathbb{ALC}$) are inconsistent with each other):

$$\overset{L_1}{\bullet_{\text{i}}} \xrightarrow{\{\langle R \geq 0.6\rangle\}} \overset{L'_2 = L'_1 \cup \{\langle B \sqcap C \leq 0.4\rangle\}}{\bullet_{\text{j}}}$$

Considering the semantics of the concept intersection in $\mathbb{ALC}$, $\langle B \sqcap C \leq 0.4\rangle$ is satisfied iff $\langle B \leq 0.4\rangle$ *or* $\langle B \leq 0.4\rangle$ are satisfied. Hence, we face a non-deterministic choice point in the completion process. Lets assume first that $\langle B \leq 0.4\rangle$ holds, then the new constraint system $L'_3 = L'_2 \cup \{\langle B \leq 0.4\rangle\}$ labeling j is unsatisfiable, since $\langle B \geq 0.6\rangle \in L'_2$. Therefore, we need to consider the second possible choice and assume that $\langle C \leq 0.4\rangle$ holds. But then again the resulting constraint system $L'_4 = L'_2 \cup \{\langle C \leq 0.4\rangle\}$ labeling j is unsatisfiable, since $\langle C \geq 0.7\rangle \in L'_2$. Since, all possible extensions lead to an unsatisfiable constraint systems node j in the completion graph, we can conclude that the given ABox $\mathcal{A}$ is unsatisfiable.

Tableau-based implementations deal with (or) non-deterministic choice points by backtracking. Since for any choice, an unsatisfiable node might be detected only after a lot of node label and graph extension steps, backtracking can become a very costly operation if the right choice amongst a set of alternatives is found only very late or the respective subgraph can not be extended to a fully-expanded and clash-free subgraph anyways.

Integration of GCIs. In order to integrate GCIs into the completion graph extension process sketched above, standard tableau-based methods (e.g. [19]) exploit the following observation: An interpretation $\mathcal{I}$ satisfies the GCI $C \sqsubseteq D$ (i.e. $\mathcal{I} \models C \sqsubseteq D$) iff for all (relevant) possibility degrees $n \in \mathsf{PossDeg}(\mathcal{K})$ and all individuals $i \in \Delta^{\mathcal{I}}$ either $\langle i : C < n\rangle$ or $\langle i : D \geq n\rangle$ holds.

To reflect this property in the tableau completion process, a non-deterministic rule for the extension of node labels is added to the inference system: for each node i in the completion graph, each GCI $C \sqsubseteq D$ in the TBox, and each (relevant) possibility degrees n, we either insert the constraint $\langle i : C < n\rangle$ or $\langle i : D \geq n\rangle$ into the current node label.

It is obvious that using this new inference rule GCIs essentially become the *major source of non-determinism* for tableau-based inference procedures: for *each* node i appearing (*somewhen*) in the completion graph, we have a distinct alternative for *each* relevant possibility degree $n \in \mathsf{PossDeg}(\mathcal{K})$ and *each* GCI $C \sqsubseteq D \in \mathcal{T}$. Hence, for each node i on a path of length l in the completion graph, we get up to $|\mathsf{PossDeg}(\mathcal{K})| \cdot |\mathcal{T}|$ different alternatives. This makes up to $l^{|\mathsf{PossDeg}(\mathcal{K})|\cdot|\mathcal{T}|}$ distinct cases that need to be considered during the completion process in the worst-case. It is further known [2] that when supporting GCIs, the maximal path length in the completion graph can be limited only by an upper bound that is (itself) exponential in the size of the input ABox $\mathcal{A}$ (using a technique called *blocking*). This gives a doubly-exponential runtime for tableau-based methods in the worst case. Consequently, tableau-based methods can spent an enormous amount of *wasted work* on backtracking even when considering a TBox $\mathcal{T}$ that includes only few GCIs and possibility degrees. However, support for GCIs is important for very many applications of DLs (and hence FDLs) in practice [19]. Without clever implementation techniques, good heuristics for guiding the backtracking search and a bit of luck with the considered problem domain, a tableau-based reasoner integration GCIs as described above might not be usable even rather small practical scenarios.

In this paper, we present a method that does not suffer from this problem, i.e. no non-deterministic choice-points are introduced in our method by considering a finite set of GCIs $\mathcal{T}$ at all.

5 A Decision Procedure based on Fuzzy Type Elimination

We present a decision procedure for KB satisfiability in $\mathbb{ALC}$ which does not rely on systematic search in the first place (as e.g. tableau-based methods), but instead constructs a canonical interpretation by means of a fixpoint construction. The so-constructed (canonical) interpretation (if non-empty) satisfies the TBox of a KB and allows to derive a model for the given knowledge base $\mathcal{K}$ iff $\mathcal{K}$ is satisfiable. In contrast to

tableau-based procedures a canonical interpretation is in general *not* tree-shaped. Further, it can be shown that the number of iterations required to reach a fixpoint is *linear* in the modal depth of $\mathcal{K}$.

Preprocessing. Without loss of generality, we can restrict ourselves to *normalized* knowledge bases [19], i.e. knowledge bases which contain only fuzzy ABox assertions of the form $\langle \alpha \geq d \rangle$, by applying the following equivalent transformation fuzzy ABox axioms: $\langle i : C \leq d \rangle \rightsquigarrow \langle i : \neg C \geq 1 - d \rangle$ and $\langle i : C = d \rangle \rightsquigarrow \langle i : C \geq d \rangle, \langle i : \neg C \geq 1 - d \rangle$. Further, we can assume that all axioms in $\mathcal{K}$ are in box normal form (BNF) [14] (i.e. the only negative concept subexpressions are of the form $\neg \forall R.C$ or negated atomic concept names $\neg C$), by exhaustively applying the following equivalent transformation to concept expressions: $\neg(C \sqcap D) \rightsquigarrow \neg C \sqcup \neg D$, $\neg(C \sqcup D) \rightsquigarrow \neg C \sqcap \neg D$, and $\neg\neg C \rightsquigarrow C$. These preprocessing steps can be performed altogether in linear time wrt. the size of the input KB.

5.1 Basic Notions and Intuition

Types. Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ denote a normalized $\mathbb{ALC}$ knowledge base in BNF. The *closure* of a knowledge base $\mathsf{cl}(\mathcal{K})$ is defined as the smallest set of concept expressions such that for all $C \in \mathsf{sub}(\mathcal{K})$, if C is not of the form $\neg D$, then $\{C, \neg C\} \subseteq \mathsf{cl}(\mathcal{K})$. Further, let $\mathsf{PossDeg}(\mathcal{K})$ denote the set of all relevant possibility degrees that can be derived from $\mathcal{K}$, i.e. $\mathsf{PossDeg}(\mathcal{K}) = \{0, 0.5, 1\} \cup \{d | \langle \alpha \geq d \rangle \in \mathcal{A}\} \cup \{1 - d | \langle \alpha \geq d \rangle \in \mathcal{A}\}$. It has been shown in [21, 22] that if $\mathcal{K}$ is satisfiable, then there is as well a model of $\mathcal{K}$ which assigns possibility degrees in $\mathsf{PossDeg}(\mathcal{K})$ only. Hence, for our purposes we do not need to consider arbitrary possibility degrees $d \in [0, 1]$, but only the *finite* set $\mathsf{PossDeg}(\mathcal{K})$ that can be derived from $\mathcal{K}$.

The closure $\mathsf{cl}(\mathcal{K})$ and the relevant possibility degrees $\mathsf{PossDeg}(\mathcal{K})$ together give us the basic vocabulary to describe individuals and their (fuzzy) properties in interpretations for $\mathcal{K}$. More specifically, the notion of a *type* allows to represent individuals of an interpretation in a syntactic way:

Definition 5.1 (Fuzzy $\mathcal{K}$-Type)**.** *A* fuzzy $\mathcal{K}$-type τ *is a maximal subset of* $\mathsf{cl}(\mathcal{K}) \times \mathsf{PossDeg}(\mathcal{K})$ *such that the following conditions are satisfied: 1. if* $\langle C, d \rangle \in \tau$ *and* $\langle C, d' \rangle \in \tau$ *then* $d = d'$ *2. if* $C = \neg C'$ *then* $\langle C, d \rangle \in \tau$ *iff* $\langle C', 1 - d \rangle \in \tau$ *3. if* $C = C' \sqcap C''$ *then* $\langle C, d \rangle \in \tau$ *iff* $\langle C', d' \rangle \in \tau$ *and* $\langle C'', d'' \rangle \in \tau$ *and* $d = min(d', d'')$ *4. if* $C = C' \sqcup C''$ *then* $\langle C, d \rangle \in \tau$ *iff* $\langle C', d' \rangle \in \tau$ *and* $\langle C'', d'' \rangle \in \tau$ *and* $d = max(d', d'')$ *5. for all* $C \sqsubseteq C' \in \mathcal{T}$*: if* $\langle C, d \rangle \in \tau$ *and* $\langle C', d' \rangle \in \tau$ *then* $d \leq d'$ *6. if* $C = \top$ *then* $\langle C, 1 \rangle \in \tau$

Since $\mathsf{cl}(\mathcal{K})$ and $\mathsf{PossDeg}(\mathcal{K})$ are both finite sets, there are at most $2^{|\mathsf{cl}(D)| \cdot |\mathsf{PossDeg}(\mathcal{K})|}$ different $\mathcal{K}$-types. Each type τ can be seen as an individual and syntactically represents *all* (fuzzy) properties that can be observed about that individual: $\langle C, d \rangle \in \tau$ represents the statement that the respective individual τ belongs to concept C with the possibility degree d. Hence, the set of all $\mathcal{K}$-types (or simply types) provides enough vocabulary to let us describe all kinds of interpretations for $\mathcal{K}$ simply be fixing how to interconnect individuals (and therefore types).

Canonical Model. It turns out that it is possible to connect types in a fixed (or canonical) way, such that the interconnection defined is consistent with *almost* all properties specified syntactically in the type. The interconnections can be derived from the types themselves:

For a set of types T we can define for each role R a *canonical accessibility relation* $\Delta_R : T \times T \rightarrow \mathsf{PossDeg}(\mathcal{K})$ that "maximally" interconnects types $\tau, \tau' \in T$ with possibility degree $d \in \mathsf{PossDeg}(\mathcal{K})$: Let $\delta(d, d') := 1$ if $d \leq d'$ and $\delta(d, d') := 1 - d$ if $d > d'$. Then, we can define Δ_R by

$$\Delta_R(\tau, \tau') := min\{\delta(d, d') | \langle \forall R.C, d \rangle \in \tau, \langle C, d' \rangle \in \tau'\}$$

if $\forall R.C \in \mathsf{cl}(\mathcal{K})$ for some $C \in \mathbf{C}$, and $\Delta_R(\tau, \tau') := 1$ otherwise.

This way, we can construct a canonical interpretation $\mathcal{I}_T$ for any given set of types T using the canonical interconnection of types by Δ_R as follows: $\mathcal{I}_T = (T, \cdot^{\mathcal{I}_T})$ with (i) for any (atomic) concept name C in $\mathcal{K}$ and any $\tau \in T$ we set $C^{\mathcal{I}_T}(\tau) = d$ if $\langle C, d\rangle \in \tau$, and (ii) $R^{\mathcal{I}_T}(\tau, \tau') = \Delta_R(\tau, \tau')$ for any role R in $\mathcal{K}$ and any $\tau, \tau' \in T$. Please note, that by our definition of $\mathcal{K}$-types, $\mathcal{I}_T$ is well-defined for any concept name or role name. However, our definition deliberately leaves open the interpretation of individuals. We therefore define in fact a class of canonical interpretations, each of which fixes a specific way of how to interpret the individuals in a KB $\mathcal{K}$.

The canonical interconnection in $\mathcal{I}_T$ is chosen in such a way that all assignments of possibility degrees to concepts of the form $C = \forall R.C \in \tau$ are lower bounds for the possibility degrees that are in fact assigned by a canonical interpretation $\mathcal{I}_T$. Hence, such a canonical interpretation is *almost* immediately a (canonical) model for the terminology $\mathcal{T}$, i.e. it satisfies that

$$C^{\mathcal{I}_T}(\tau) = d \text{ iff } \langle C, d\rangle \in \tau \qquad (*)$$

for *almost* all $C \in \mathsf{cl}(\mathcal{K})$ and therefore $\mathcal{I}_T \models C \sqsubseteq C'$ for all $C \sqsubseteq C' \in \mathcal{T}$ by clause (5) in our definition of $\mathcal{K}$-types. That $(*)$ is satisfied by $\mathcal{I}_T$ is straightforward for the cases of concept names C, or complex concepts of the form $C = C' \sqcap C''$, $C = C' \sqcup C''$, $C = \neg C'$ and the $C^{\mathcal{I}_T}(\tau) \geq d$ case for $C = \forall R.C$ by our definition of types and the definition of Δ_R. The only cases where $(*)$ can be violated by $\mathcal{I}_T$ is for types τ containing universally role restricted concepts $\forall R.C$ that are assigned a possibility degree which is *too small* (wrt. the R-successor types τ' in $\mathcal{I}_T$) to properly reflect the semantics of $\forall R.C$ in $\mathbb{ALC}$, i.e. to coincide with the *greatest* lower bound of the set

$$\{max(1 - R^{\mathcal{I}_T}(\tau, \tau'), C^{\mathcal{I}_T}(\tau')) \,|\, \tau' \in T\}$$

Types τ in which the possibility degree assigned d to $\forall R.C$ is too small to be consistent with the semantics of $\mathbb{ALC}$ are called *bad types*. Bad types $\tau \in T$ can be detected easily, since they satisfy that there exist $R \in \mathbf{R}, C \in \mathcal{C}(\Sigma), d \in \mathsf{PossDeg}(\mathcal{K})$ s.t. $\langle \forall R.C, d\rangle \in \tau$ and for all $\tau' \in T$: if $\langle C, d'\rangle \in \tau'$ then $max(1 - \Delta_R(\tau, \tau'), d') > d$.

This suggests the following simple algorithm (which uses a *fuzzy type elimination* process as its core): in order to compute a maximal interpretation that satisfies all terminological axioms, we start off with the maximal set of types (i.e all $\mathcal{K}$-types) and iteratively fix all problems that prevent $(*)$ from being satisfied by removing bad types. This way, we must eventually reach a fixpoint after finitely many steps. If the resulting set of types is non-empty, we know that $(*)$ must hold (since all problems have been fixed) and therefore we can be certain that the corresponding canonical interpretation satisfies $\mathcal{T}$ (and covers all other possible models of $\mathcal{T}$ at the same time). Hence, we eventually need to check if all ABox axioms are satisfied by the canonical interpretation. If this is the case, we have found a model for $\mathcal{K}$, otherwise, we know that there can not be any interpretation that satisfies both $\mathcal{T}$ and $\mathcal{A}$ at the same time. In other words, $\mathcal{K}$ is not satisfiable.

5.2 Algorithm

The type elimination process sketched above can be formalized as shown in Fig.1. Note that the emptiness test for the fixpoint T is covered implicitly: if the fixpoint T is empty, then the test in the if-statement fails trivially.

5.3 Soundness, Completeness and Termination

The termination, soundness, and completeness of our algorithm can be proven formally.

procedure satisfiable($\mathcal{K}$): boolean
T := $\{\tau | \tau$ is a $\mathcal{K}$-type $\}$;
repeat
  $T' := T$;
  $T := T' \setminus badtypes(T')$;
until $T = T'$;
if *there exists a total function* $\pi : Ind_{\mathcal{A}} \rightarrow T$ *s.t.* $\langle C, d' \rangle \in \pi(o)$ *and* $d \leq d'$ *for each* $\langle o : C \geq d \rangle \in \mathcal{A}$, *and* $\Delta_R(\pi(o), \pi(o')) \geq d$ *for each* $\langle R(o, o') \geq d \rangle \in \mathcal{A}$ **then**
  return true;
end
return false;

function $badtypes(T) : 2^T$
return $\{\tau \in T | \langle \forall R.C, d \rangle \in \tau$ and for all $\tau' \in T$: if $\langle C, d' \rangle \in \tau'$ then $max(1 - \Delta_R(\tau, \tau'), d') > d\}$;

Algorithm 1: The Type Elimination-based Decision Procedure $\mathbf{FixIt}(\mathbb{ALC})$

Theorem 5.1 (Termination)**.** *For any* $\mathbb{ALC}$ *knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *the algorithm* $\mathbf{FixIt}(\mathbb{ALC})$ *terminates after finitely many steps with either* true *or* false *as return value.*

Proof. The initialization step of the algorithm takes finitely many steps since the number of $\mathcal{K}$-types is finite. The repeat-loop must terminate after finitely many steps, since we start with a finite set of types T in the beginning: if we do not remove any type in an iteration (i.e. $badtypes(T') = \emptyset$) we have $T = T'$ at the end of the loop (i.e. reaching a fixpoint) and therefore terminate the loop. On the other hand, if $badtypes(T') \neq \emptyset$ in an iteration, at least one type is removed from T' and hence $T \subset T'$. This means, that the input set of types T for the next iteration is finite and strictly smaller. Clearly, the empty set is a fixpoint of $badtypes(\cdot)$ too, i.e. $badtypes(\emptyset) = \emptyset$. Hence, we can repeat the loop only finitely many times until we finally will reach a fixpoint. Since this fixpoint T is a subset of the finite set of the initial set of types and there are only finitely many possible mappings π to consider, deciding the criterion in the if-statement (based on T) takes as well only finitely many steps. Therefore, the algorithm terminates with one of the return-statements that give as a result either *true* or *false*. □

The following lemma is a key element of the soundness and completeness proof and shows that by successively removing bad types we can indeed ensure that types encode possibility degree assignments to concepts that coincide with the canonical interpretation, and that any such canonical interpretation is a model of the $\mathcal{T}$.

Let T be the set of types that is computed as the fixpoint in the algorithm $\mathbf{FixIt}(\mathbb{ALC})$, i.e. $badtypes(T) = \emptyset$ and let $\mathcal{I}_T = (T, \cdot^{\mathcal{I}_T})$ be a canonical interpretation for T as defined above.

Lemma 5.2. *For each* $\mathcal{K}$*-type* τ*, concept* $C \in \mathsf{cl}(\mathcal{K})$ *and* $d \in \mathsf{PossDeg}(\mathcal{K})$ *it holds that* $C^{\mathcal{I}_T}(\tau) = d$ *iff* $\langle C, d \rangle \in \tau$*. Further,* $\mathcal{I}_T \models \mathcal{T}$.

Proof. For the first part of the lemma, let τ be any $\mathcal{K}$-type and $d \in \mathsf{PossDeg}(\mathcal{K})$ be any relevant possibility degree.

We show by induction over the structure of concepts $C \in \mathsf{cl}(\mathcal{K})$ that $\langle C, d\rangle \in \tau$ iff $C^{\mathcal{I}_T}(\tau) = d$: the base case (i.e. $C \in \mathsf{cl}(\mathcal{K})$ is an atomic concept name $C \in \mathbf{C}$ or $C = \top$) is trivially satisfied by our definition of $\mathcal{I}_T$. For the induction step, we consider the different cases of compound concept expressions $C \in \mathsf{cl}(\mathcal{K})$ one-by-one:

1. for $C = C_1 \sqcap C_2 \in \mathsf{cl}(\mathcal{K})$, we know that $C_1, C_2 \in \mathsf{cl}(\mathcal{K})$. By clause (3) in our definition of types, we know that $\langle C, d\rangle = \langle C_1 \sqcap C_2, d\rangle \in \tau$ iff $\langle C_1, d_1\rangle \in \tau$ and $\langle C_2, d_2\rangle \in \tau$ and $d = min(d_1, d_2)$. Applying the induction hypothesis to C_1 and C_2, we know that this is the case iff ${C_1}^{\mathcal{I}_T}(\tau) = d_1$ and ${C_2}^{\mathcal{I}_T}(\tau) = d_2$ and $d = min(d_1, d_2)$ iff $d = min({C_1}^{\mathcal{I}_T}(\tau), {C_2}^{\mathcal{I}_T}(\tau))$ iff $d = (C_1 \sqcap C_2)^{\mathcal{I}_T} = C^{\mathcal{I}_T}$ by the by the semantics of $\sqcap$ in $\mathbb{ALC}$.

2. for $C = C_1 \sqcup C_2 \in \mathsf{cl}(\mathcal{K})$ the proof is analogous.

3. for $C = \neg D \in \mathsf{cl}(\mathcal{K})$, we know that $D \in \mathsf{cl}(\mathcal{K})$ by the definition of $\mathsf{cl}(\mathcal{K})$. Because of clause (2) and the maximality requirement in our definition of $\mathcal{K}$-types, we know that $\langle C, d\rangle = \langle \neg D, d\rangle \in \tau$ iff $\langle D, 1-d\rangle \in \tau$. Applying the induction hypothesis for D, we know that this holds iff $D^{\mathcal{I}_T}(\tau) = 1 - d$ iff $(\neg D)^{\mathcal{I}_T}(\tau) = C^{\mathcal{I}_T}(\tau) = d$ by the semantics of concept negation in $\mathbb{ALC}$.

4. for $C = \forall R.D \in \mathsf{cl}(\mathcal{K})$, $D \in sub(\mathcal{K})$ holds and hence $D \in \mathsf{cl}(\mathcal{K})$ by the definition of $\mathsf{cl}(\mathcal{K})$.

First, we show one direction, i.e. that $C^{\mathcal{I}_T}(\tau) = d$ if $\langle C, d\rangle \in \tau$: Assume that $\langle C, d\rangle = \langle \forall R.D, d\rangle \in \tau$. According to the semantics of the universal role restriction in $\mathbb{ALC}$ and our definition of $\mathcal{I}_T$, we have $C^{\mathcal{I}_T}(\tau) = (\forall R.D)^{\mathcal{I}_T}(\tau) = inf_{\tau' \in T}\{max(1-R^{\mathcal{I}_T}(\tau, \tau'), D^{\mathcal{I}_T}(\tau'))\} = inf_{\tau' \in T}\{max(1-\Delta_R(\tau, \tau'), D^{\mathcal{I}_T}(\tau'))\}$. We show that d is a lower bound for $\{max(1-\Delta_R(\tau, \tau'), D^{\mathcal{I}_T}(\tau'))|\tau' \in T\}$: Assume there exists a $\tau' \in T$ s.t. $d > max(1-\Delta_R(\tau, \tau'), D^{\mathcal{I}_T}(\tau'))$. Let $D^{\mathcal{I}_T}(\tau') = d'$. Applying the induction hypothesis to $D \in \mathsf{cl}(\mathcal{K})$, we know $\langle D, d'\rangle \in \tau'$. Hence, both $d > 1 - \Delta_R(\tau, \tau')$ and $d > d'$ must hold. Hence $\Delta_R(\tau, \tau') > 1 - d$. But, since $\langle \forall R.D, d\rangle \in \tau$ this is not possible by our by our definition of Δ_R, because $\Delta_R(\tau, \tau') \leq 1 - d$. From the contradiction we can conclude that d is in fact a lower bound for the considered set. Therefore, $d \leq inf_{\tau' \in T}\{max(1 - \Delta_R(\tau, \tau'), D^{\mathcal{I}_T}(\tau'))\}$

Next, we show that $inf_{\tau' \in T}\{max(1 - \Delta_R(\tau, \tau'), D^{\mathcal{I}_T}(\tau'))\} \leq d$ too, by proving that there exists a $\tau' \in T$ s.t. $max(1 - \Delta_R(\tau, \tau'), D^{\mathcal{I}_T}(\tau')) \leq d$. Assume, the contrary, i.e. for all $\tau' \in T$: $max(1 - \Delta_R(\tau, \tau'), D^{\mathcal{I}_T}(\tau')) > d$ (‡). By applying our induction hypothesis to D and τ', we know that this is the case iff for all $\tau' \in T$: if $\langle D, d'\rangle \in \tau'$ then $max(1 - \Delta_R(\tau, \tau'), d') > d$. But then, τ would be a bad type which contradicts the fact that T the computed fixpoint which can not contain any bad types (i.e. $\tau \in badtypes(T) = \emptyset$). Hence our assumption (‡) must be wrong, and we can conclude that $inf_{\tau' \in T}\{max(1 - \Delta_R(\tau, \tau'), D^{\mathcal{I}_T}(\tau'))\} \leq d$. Therefore, $d = inf_{\tau' \in T}\{max(1 - \Delta_R(\tau, \tau'), D^{\mathcal{I}_T}(\tau'))\}$, and hence $d = C^{\mathcal{I}_T}(\tau)$.

The other direction of the induction hypothesis (i.e. that $\langle \forall R.D, d\rangle \in \tau$ if $(\forall R.D)^{\mathcal{I}_T}(\tau) = d$) can now be proven as follows: Assume that $(\forall R.D)^{\mathcal{I}_T}(\tau) = d$ (†) but $\langle \forall R.D, d\rangle \notin \tau$. By the maximality requirement in our definition of $\mathcal{K}$-types there must hence exist a $d \in \mathsf{PossDeg}(\mathcal{K})$ s.t. $\langle \forall R.D, d'\rangle \in \tau$ and $d' \neq d$. Using the same argument as for the if-direction in this case, we can therefore conclude that $(\forall R.D)^{\mathcal{I}_T}(\tau) = d' \neq d$ which contradicts (†). Hence, our assumption must be wrong and $\langle \forall R.D, d\rangle \in \tau$ must hold whenever $(\forall R.D)^{\mathcal{I}_T}(\tau) = d$.

For the second part of the lemma, to show that $\mathcal{I}_T \models \mathcal{T}$, assume that for some $\alpha = C \sqsubseteq C' \in \mathcal{T}$ and some $\tau \in T$ it holds that $C^{\mathcal{I}_T}(\tau) > C'^{\mathcal{I}_T}(\tau)$, in other words, if $C^{\mathcal{I}_T}(\tau) = d$ and $C'^{\mathcal{I}_T}(\tau) = d'$ then $d > d'$. Thus, we can deduce (from the first part of this lemma) that, if $\langle C, d\rangle \in \tau$ and $\langle C', d'\rangle \in \tau$ then $d > d'$. However, by our definition of $\mathcal{K}$-type (i.e. clause (5)), we also know that in this case $d \leq d$ must hold, which is contradictive. Hence, our assumption must be wrong and $\mathcal{I}_T \models \alpha$ for each $\alpha \in \mathcal{T}$ which means that $\mathcal{I}_T \models \mathcal{T}$.

□

Theorem 5.3 (Soundness)**.** *If* $\mathbf{FixIt}(\mathbb{ALC})$ *returns true for a* $\mathbb{ALC}$ *knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$*, then* $\mathcal{K}$ *is satisfiable.*

Proof. We show that a canonical interpretation $\mathcal{I}_T$ for the computed fixpoint T can be extended to a model of $\mathcal{K}$. By Lemma 5.2, we already know that $\mathcal{I}_T \models \mathcal{T}$. We now show, that $\mathcal{I}_T$ can be extended such that $\mathcal{I}_T \models \mathcal{A}$ too, which completes the proof: Since the algorithm returns true, there exist a total function $\pi : Ind_{\mathcal{A}} \rightarrow T$ s.t. $\langle C, d' \rangle \in \pi(o)$ and $d \leq d'$ for each $\langle o : C \geq d \rangle \in \mathcal{A}$ ($\star$), and $\Delta_R(\pi(o), \pi(o')) \geq d$ for each $\langle R(o, o') \geq d \rangle \in \mathcal{A}$ (†). We extend the definition of $\mathcal{I}_T$ to the ABox $\mathcal{A}$ as follows: for all Abox individual names $o \in Ind_{\mathcal{A}}$, we set $o^{\mathcal{I}_T} := \pi(o) \in T$. First, consider an ABox axiom of the form $\alpha = \langle o : C \geq d \rangle \in \mathcal{A}$. Then, $\mathcal{I}_T \models \alpha$ iff $C^{\mathcal{I}_T}(o^{\mathcal{I}_T}) \geq d$, iff $C^{\mathcal{I}_T}(\pi(o)) \geq d$ iff there exists a $d' \geq d$ s.t. $C^{\mathcal{I}_T}(\pi(o)) = d'$. By Lemma 5.2 this is the case iff there exists a $d' \geq d$ s.t. $\langle C, d' \rangle \in \pi(o)$ which is satisfied since ($\star$) holds. Second, consider an ABox axiom of the form $\alpha = \langle R(o, o') \geq d \rangle \in \mathcal{A}$. Then, $\mathcal{I}_T \models \alpha$ iff $R^{\mathcal{I}_T}(o^{\mathcal{I}_T}, o'^{\mathcal{I}_T}) \geq d$ iff $\Delta_R(\pi(o), \pi(o')) \geq d$ (by Def. of the extended $\mathcal{I}_T$). The latter is satisfied because of (†). □

A second key element for the completeness proof is the following lemma that shows that our canonical way of interconnecting types (in the fixpoint set) is maximal or the strongest possible one in the following sense: the interconnection R of individuals o, o' defined by any model $\mathcal{I}$ of $\mathcal{K}$ is covered by the canonical interconnection Δ_R of the respective types $\tau(o), \tau(o')$ representing o, o' in $\mathcal{I}$.

Lemma 5.4. *Let* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ *be any model of* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$*. For each individual* $o \in \Delta^{\mathcal{I}}$ *we define its corresponding type* $\tau(o) := \{\langle C, d \rangle \in \mathsf{cl}(\mathcal{K}) \times \mathsf{PossDeg}(\mathcal{K}) | C^{\mathcal{I}}(o) = d\}$*. Then,* $\Delta_R(\tau(o), \tau(o')) \geq R^{\mathcal{I}}(o, o')$ *for all* $o, o' \in \Delta^{\mathcal{I}}$*.*

Proof. Assume that there exist $o, o' \in \Delta^{\mathcal{I}}$ s.t. $\Delta_R(\tau(o), \tau(o')) < R^{\mathcal{I}}(o, o')$. By our definition of Δ_R, we then know that $\delta(d, d') < R^{\mathcal{I}}(o, o')$ ($*$) for some $\langle \forall R.C, d \rangle \in \tau(o)$ and $\langle C, d' \rangle \in \tau(o')$. From the definition of $\tau(\cdot)$, we know that $\delta(d, d') < R^{\mathcal{I}}(o, o')$ for some $o, o' \in \Delta^{\mathcal{I}}$ s.t. $(\forall R.C)^{\mathcal{I}}(o) = d$ and $C^{\mathcal{I}}(o') = d'$. From the semantics of $\forall R.C$ in $\mathbb{ALC}$, we derive $d = inf_{o'' \in \Delta^{\mathcal{I}}}\{max(1 - R^{\mathcal{I}}(o, o''), C^{\mathcal{I}_T}(o''))\}$. Hence, in particular $d \leq max(1 - R^{\mathcal{I}}(o, o'), C^{\mathcal{I}_T}(o')) = max(1 - R^{\mathcal{I}}(o, o'), d')$ (†). We consider two cases: first, $d' < d$, then in order to satisfy (†), $max(1 - R^{\mathcal{I}}(o, o'), d') = 1 - R^{\mathcal{I}}(o, o')$ must hold and (†) simplifies to $d \leq 1 - R^{\mathcal{I}}(o, o')$ iff $R^{\mathcal{I}}(o, o') \leq 1 - d$. Since $d' < d$, ($*$) simplifies to $1 - d < R^{\mathcal{I}}(o, o')$, hence, $1 - d < R^{\mathcal{I}}(o, o') \leq 1 - d$ which is contradictory. In the second case, we assume that $d' \geq d$. Hence, $\delta(d, d') = 1$. Then, (†) simplifies to $1 < R^{\mathcal{I}}(o, o')$, which is contradictory, since $R^{\mathcal{I}}(o, o') \in \mathsf{PossDeg}(\mathcal{K})$ and 1 is the maximum possibility degree in $\mathsf{PossDeg}(\mathcal{K})$. Therefore, in both cases we reach a contradiction and can conclude that our assumption must be wrong. This concludes the proof.

□

Theorem 5.5 (Completeness)**.** *If an* $\mathbb{ALC}$ *knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *is satisfiable, then* $\mathbf{FixIt}(\mathbb{ALC})$ *returns true for* $\mathcal{K}$

Proof. Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be any model of $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, i.e. $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$. In [21, 22] it is shown that a KB in $\mathbb{ALC}$ is consistent iff there is a model $\mathcal{I}$ of the KB which only assigns possibility degrees that occur in the ABox $\mathcal{A}$ for interpreting atomic concept or role names. Hence, without loss of generality, we can assume in the following that $\mathcal{I}$ assigns possibility degrees in $\mathsf{PossDeg}(\mathcal{K})$ only.

For each individual $o \in \Delta^{\mathcal{I}}$ we define its corresponding type $\tau(o) := \{\langle C, d \rangle \in \mathsf{cl}(\mathcal{K}) \times \mathsf{PossDeg}(\mathcal{K}) | C^{\mathcal{I}}(o) = d\}$ and define $T_{\mathcal{I}} := \{\tau(o) | o \in \Delta^{\mathcal{I}}\}$. It is easy to see that $T_{\mathcal{I}}$ is a set of $\mathcal{K}$-types. Further, $T_{\mathcal{I}} \neq \emptyset$ since $\Delta^{\mathcal{I}} \neq \emptyset$.

Let $T^{(i)}$ denote the set of types that is computed after i iterations of the repeat-loop in our algorithm. We first show that $T_{\mathcal{I}} \subseteq T^{(i)}$ for all $i \geq 0$ by induction over the number of iterations i:

In the base case $i = 0$, our initialization step sets $T^{(0)}$ to contain all $\mathcal{K}$-types. Since $T_{\mathcal{I}}$ consists of $\mathcal{K}$-types only, $T_{\mathcal{I}} \subseteq T^{(0)}$ must hold. To proof the induction step, we assume that $T_{\mathcal{I}} \subseteq T^{(i)}$ but $T_{\mathcal{I}} \not\subseteq T^{(i+1)}$. Therefore, there must be a $\tau(o) \in T_{\mathcal{I}}$ s.t. $\tau(o) \in T^{(i)}$ but not $\tau(o) \in T^{(i+1)}$. From the repeat-loop in the algorithm, we know that $T^{(i+1)} = T^{(i)}\ badtypes(T^{(i)})$. Consequently, $\tau(o)$ must be a bad type $\tau(o) \in badtypes(T^{(i)})$ and we can not have reached a fix-point yet.

From our definition of bad-types we can derive that there must exist a $\langle \forall R.C, d\rangle \in \tau(o)$ and for all $\tau' \in T^{(i)}$: if $\langle C, d'\rangle \in \tau'$ then $max(1 - \Delta_R(\tau, \tau'), d') > d\}$ (‡). Since $T_{\mathcal{I}} \subseteq T^{(i)}$ (‡) must hold in particular for all $\tau(o') \in T_{\mathcal{I}}$. Using our definition of $\tau(\cdot)$ we can rephrase (‡) as follows: there must exist a $\forall R.C \in \mathsf{cl}(\mathcal{K})$ s.t. for all $o' \in \Delta^{\mathcal{I}}$: $max(1 - \Delta_R(\tau(o), \tau(o')), C^{\mathcal{I}}(o')) > (\forall R.C)^{\mathcal{I}}(o)$ ($\star$).

By Lemma 5.4, we know that $\Delta_R(\tau(o), \tau(o')) \geq R^{\mathcal{I}}(o, o')$ for all $o, o' \in \Delta^{\mathcal{I}}$. Hence, $1-\Delta_R(\tau(o), \tau(o')) \leq 1 - R^{\mathcal{I}}(o, o')$ ($*$) for all $o, o' \in \Delta^{\mathcal{I}}$. Since $max(a, b) \leq max(a', b)$ for any a, a', b s.t. $a \leq a'$, we can reformulate ($\star$) using ($*$) as follows: there must exist a $\forall R.C \in \mathsf{cl}(\mathcal{K})$ s.t. for all $o' \in \Delta^{\mathcal{I}}$: $max(1 - R^{\mathcal{I}}(o, o'), C^{\mathcal{I}}(o')) > (\forall R.C)^{\mathcal{I}}(o)\}$ (♮), which contradicts the fact that $(\forall R.C)^{\mathcal{I}}(o) = inf_{o' \in \Delta^{\mathcal{I}}}\{max(1 - R^{\mathcal{I}}(o, o'), C^{\mathcal{I}}(o'))\}$: Since $R^{\mathcal{I}}(o, o') \in \mathsf{PossDeg}(\mathcal{K})$ and $C^{\mathcal{I}}(o') \in \mathsf{PossDeg}(\mathcal{K})$, we know that $max(1 - R^{\mathcal{I}}(o, o'), C^{\mathcal{I}}(o')) \in \mathsf{PossDeg}(\mathcal{K})$ for all $o' \in \Delta^{\mathcal{I}}$ by our definition of $\mathsf{PossDeg}(\mathcal{K})$. Because $\Delta^{\mathcal{I}} \neq \emptyset$ and $\mathsf{PossDeg}(\mathcal{K})$ is a finite set, there must exist an $o^* \in \Delta^{\mathcal{I}}$ for which $max(1 - R^{\mathcal{I}}(o, o^*), C^{\mathcal{I}}(o^*))$ is minimal, i.e. $max(1 - R^{\mathcal{I}}(o, o^*), C^{\mathcal{I}}(o^*)) \leq max(1 - R^{\mathcal{I}}(o, o'), C^{\mathcal{I}}(o'))$ for all $o' \in \Delta^{\mathcal{I}}$. Hence, $d^* := max(1 - R^{\mathcal{I}}(o, o^*), C^{\mathcal{I}}(o^*)) \in \mathsf{PossDeg}(\mathcal{K})$ is a lower bound for the set $\{max(1 - R^{\mathcal{I}}(o, o'), C^{\mathcal{I}}(o') | o' \in \Delta^{\mathcal{I}}\}$. However, from (♮) we know that $d < d^*$, hence d can not be the greatest lower bound (i.e. the infimum) of the set $\{max(1 - R^{\mathcal{I}}(o, o'), C^{\mathcal{I}}(o') | o' \in \Delta^{\mathcal{I}}\}$, hence $\forall R.C)^{\mathcal{I}}(o) = d^* > d$ which is contradictive.

Therefore, our assumption that $\tau(o)$ is a bad type must be wrong and we have completed the proof of the induction step as well as the induction argument.

We continue the proof of the lemma as follows: since $T = T^{(i)}$ is the fixpoint that is computed in the loop in our algorithm in i steps for some $i \geq 0$, we know that $T_{\mathcal{I}} \subseteq T^{(i)} = T$. Consider the mapping $\pi_{\mathcal{I}} : Ind_{\mathcal{A}} \to T$ defined by $\pi_{\mathcal{I}}(o) := \tau(o^{\mathcal{I}})$ for all $o \in Ind_{\mathcal{A}}$. Then, $\pi_{\mathcal{I}}$ is a well-defined, total function from $Ind_{\mathcal{A}}$ to T. We now show that this specific mapping $\pi_{\mathcal{I}}$ satisfies the condition that is checked in the if-statement of the algorithm:

In the first case, we consider any Abox axiom $\alpha \in \mathcal{A}$ of the form $\alpha = \langle o : C \geq d\rangle$. Since $\mathcal{I} \models \mathcal{A}$, $\mathcal{I} \models \alpha$ must hold. $\mathcal{I} \models \alpha$ iff $C^{\mathcal{I}}(o^{\mathcal{I}}) \geq d$ iff $C^{\mathcal{I}}(o^{\mathcal{I}}) = d'$ for some $d' \in \mathsf{PossDeg}(\mathcal{K})$ with $d' \geq d$ iff $\langle C, d\rangle \in \tau(o^{\mathcal{I}})$ for some $d' \in \mathsf{PossDeg}(\mathcal{K})$ with $d' \geq d$ (by Lemma 5.2) iff $\langle C, d\rangle \in \pi_{\mathcal{I}}(o)$ for some $d' \in \mathsf{PossDeg}(\mathcal{K})$ with $d' \geq d$ (by our definition of $\pi_{\mathcal{I}}$). Hence, the respective part of the if-condition for α holds for $\pi_{\mathcal{I}}$. In the second case, we consider any Abox axiom $\alpha \in \mathcal{A}$ of the form $\alpha = \langle R(o, o') \geq d\rangle$. Since, $\mathcal{I} \models \mathcal{A}, \mathcal{I} \models \alpha$ must hold. $\mathcal{I} \models \alpha$ holds iff $R^{\mathcal{I}}(o^{\mathcal{I}}, o'^{\mathcal{I}}) \geq d$. Since $\Delta_R(\tau(o), \tau(o')) \geq R^{\mathcal{I}}(o^{\mathcal{I}}, o'^{\mathcal{I}})$ by Lemma 5.4, we know that $\Delta_R(\tau(o^{\mathcal{I}}), \tau(o'^{\mathcal{I}})) \geq d$ and therefore $\Delta_R(\pi_{\mathcal{I}}(o), \pi_{\mathcal{I}}(o)) \geq d$ by our definition of $\pi_{\mathcal{I}}$. Hence, the respective part of the if-condition for α is as well satisfied by $\pi_{\mathcal{I}}$. Consequently, the tested if-condition is satisfied by $\pi_{\mathcal{I}}$ and the algorithm returns *true*. □

This leads to the main result, which is an immediate consequence of Theorems 5.3, 5.5, and 5.1:

Corollary 5.6. *The algorithm* $\mathbf{FixIt}(\mathbb{ALC})$ *is a sound and complete decision procedure for knowledge base satisfiability in* $\mathbb{ALC}$.

5.4 Runtime and Space Requirements

We now analyze the runtime and space requirements of our algorithm based on a naive implementation model to derive upper bounds. The considered implementation works directly on types and explicit representation of sets of types.

In the following we assume that the number p of different possibility degrees that can occur in any KB $\mathcal{K}$ is fixed. This assumption seems realistic and does not restrict applications of FDLs in practice, i.e. we can assume a limited (numerical) resolution of sensors and algorithms (or humans) assigning possibility degrees to individual observations. Further, we consider the the computation of the basic functions $min, max, 1-d$ and comparisons $d \leq d'$ as atomic operations with unit costs.

Representation. According to Def. 5.1, a $\mathcal{K}$-type τ is a function $\tau : \mathsf{cl}(\mathcal{K}) \rightarrow \mathsf{PossDeg}(\mathcal{K})$ that satisfies a particular consistency property. Fix some total order $\langle C_1, C_2, \ldots, C_k \rangle$ on the subset $\mathsf{cl}_+(\mathcal{K})$ of $\mathsf{cl}(\mathcal{K})$ that consists of all positive concept expression $C_i \in \mathsf{cl}(\mathcal{K})$. Then, we can represent a type τ by a sequence of pairs

$$\tau = \langle d_1, \overline{d_1} \rangle, \langle d_2, \overline{d_2} \rangle, \ldots, \langle d_k, \overline{d_k} \rangle$$

with $k = |\mathsf{cl}_+(\mathcal{K})| \in O(|\mathcal{K}|)$, where d_i represents the possibility degree assigned to a positive concept subexpression $C_i \in \mathsf{cl}(\mathcal{K})$ and $\overline{d_i}$ represents the possibility degree assigned to a negative concept subexpression $\neg C_i \in \mathsf{cl}(\mathcal{K})$. There are only finitely many relevant possibility degrees. Assuming an binary encoding $\langle \cdot \rangle_{bin} : \mathsf{PossDeg}(\mathcal{K}) \rightarrow \{0,1\}^p$ of possibility degrees, each such sequence requires at most $2 \cdot k \cdot log_2(p) \in O(k)$ bits. Clearly, not each such sequence represents a $\mathcal{K}$-type. However, for any sequence checking the consistency requirements from Def. 5.1 can be done time $O(k \cdot |\mathcal{T}|)$ and space $O(1)$: property (1) is satisfied already by our encoding, properties (2)-(3) and (6) can each be decided in $O(k)$, property (5) requires $O(k \cdot |\mathcal{T}|)$ computation steps. In any case, we need only $O(1)$ additional space for the computation.

The algorithm can be separated into three different phases: the initialization step, the fixpoint computation, and the final test. We analyze all of these steps one-by-one.

Initialization Phase. To compute the set of all $\mathcal{K}$-types, we generate any sequence $\langle d_1, \overline{d_1} \rangle, \langle d_2, \overline{d_2} \rangle, \ldots, \langle d_k, \overline{d_k} \rangle$ s.t. $d_i, \overline{d_k} \in \mathsf{PossDeg}(\mathcal{K})$. For each sequence this requires at most $2k$ steps. Testing if such a generated sequence satisfies the consistency requirements, takes at most $O(k \cdot |\mathcal{T}|)$ steps and only constant additional space. Any sequence that satisfies the consistency requirements is stored in a linked list. During the initialization phase, we need to check p^{2k} sequence, which requires $O(p^{2k} \cdot k \cdot |\mathcal{T}|)$ computation steps and $O(p^{2k} \cdot k)$ space.

Fixpoint Computation. The current set of types T in the fixpoint computation is represented as linked list from which elements are removed during this phase. Clearly, the function $\delta(d, d')$ can be computed in constant time. The computation of $\Delta_R(\tau, \tau')$ therefore can be performed in $O(k^2)$ computation steps and constant additional space for any two given types τ, τ'. Given a type $\tau \in T$, we can then determine if τ is a bad type (wrt. T) in $O(k \cdot (|T| \cdot k^2)) = O(k^3 \cdot |T|)$ basic computation steps using constant space only. This test has to be performed for all $|T|$ types in T in order to compute T'. Removing a bad type from the current list of types T requires $O(|T|)$ time and constant additional space. To find out if we reached a fixpoint after an iteration, we can simply use a boolean variable, which is set to $false$ at the beginning of each loop and set to $true$ if a type is removed from the list. This requires at most $O(|T|)$ additional assignments and one

boolean comparison during each iteration and constant additional space. Hence, each iteration of the loop (to compute T') requires $O((k^3+1)\cdot|T|^2+|T|)$ computation steps and constant additional space.

Let t denote the size of the initial set of types T. Since we need at most t iterations to reach a fixpoint, we can compute the fixpoint in at most $O((k^3+1)\cdot t^3+t)$ steps using $O(t)$ additional memory units. Since $t \leq p^{2k}$, we can derive $O((k^3+1)\cdot p^{6k}+p^{2k})$ as an upper bound on the number of computation steps performed for the fixpoint computation and $O(p^{2k})$ as an upper bound for the additionally required space.

Final Test. We can represent a total total mapping $\pi : Ind_{\mathcal{A}} \rightarrow T$ as a mapping from $Ind_{\mathcal{A}}$ to the index of an element in the list T. This requires, $O(i \cdot log_2(t))$ additional memory units with $i = |Ind_{\mathcal{A}}|$ and lookups for values of π can be performed in $O(t)$ time using a hashing function and a subsequent iteration over the list representing T. Given such a mapping π, we can therefore determine if the required property in the if-statement in Algorithm 1 is satisfied by π in at most $O(a\cdot(t\cdot k^2))$ steps (where $a=|\mathcal{A}|$) using $O(1)$ additional memory only. We can generate any such mapping $\pi : Ind_{\mathcal{A}} \rightarrow T$ in at most $O(i\cdot log_2(t))$ steps. Further, there are at most t^i different such mapping, which we can generate and test one-by-one. Hence, we can implement that final test using at most $O(t^i\cdot(i\cdot log_2(t)+a\cdot(t\cdot k^2))) = O(p^{2ki}\cdot(i\cdot k\cdot log_2(p)+a\cdot(p^{2k}\cdot k^2)))$ steps and $O(i\cdot k\cdot log_2(p))$ additional memory units.

Overall runtime and space bounds. For an upper bound on the overall execution of the algorithm, we can sum up the runtime and space bounds for the three different phases above: In regard of runtime, the algorithm requires no more than $O(p^{2k}\cdot k\cdot|\mathcal{T}|+(k^3+1)\cdot p^{6k}+p^{2k}+(p^{2ki}\cdot(i\cdot k\cdot log_2(p)+a\cdot(p^{2k}\cdot k^2))))$ computation steps. In regard of the space, the algorithm requires no more than $O(p^{2k}\cdot k+p^{2k}+i\cdot k\cdot log_2(p)+|\mathcal{K}|)$ space. Since $k, i \in O(|\mathcal{K}|)$ and since we assume that p is a constant (and hence independent from a size of a KB $\mathcal{K}$), the algorithm requires at most exponentially many computations steps and an exponential amount of memory in regard of the size of the input KB $\mathcal{K}$.

This means that (under the realistic assumptions) our algorithm can be implemented in a way that is *worst-case optimal* in regard of the runtime of the algorithm, since by Theorem 3.5 the problem of determining the satisfiability of a KB in $\mathbb{ALC}$ requires exponential time (wrt. the size of the input KB) in the worst case. Further, note that applying a standard tableau-based decision procedure (e.g. [2]) to (crisp) equisatisfiable $\mathcal{ALC}$-reduction of a general $\mathbb{ALC}$ KB usually yields only a NEXPTIME-upper-bound on the runtime of such an (indirect) decision procedure. Therefore, indirect reasoning approaches by reduction to classical tableau-based methods usually can only give *substantially worse* runtime guarantees.

Consequently, the major obstacles when using the algorithm in practice are (i) the exponential space requirement and (ii) the necessity to consider all types in $\tau \in T$ during each loop one-by-one. These problems are mainly based on the fact that we use *explicit* representations of set of types. A potential solution to these problems is known from the area of Symbolic Model Checking [12] and has already been successfully applied for the implementation of the $\mathcal{KBDD}$ procedure in [14]: the use *implicit* (or declarative) representations of sets of types by means of formulae and to implement set-theoretic operations by formula modifications and (un)satisfiability tests. In particular, set-theoretic operations on sets can be implemented as set-at-a-time operations instead of element-at-a-time operations which can speed up computation significantly.

6 Related Work

Our method $\mathbf{FixIt}(\mathbb{ALC})$ generalizes the principle (i.e. a type elimination process) underlying the top-down variant of the $\mathcal{KBDD}$ procedure proposed in [14] for the modal logic **K** to the (more expressive) FDL $\mathbb{ALC}$. Further, our method integrates (fuzzy) ABoxes and TBoxes in the inference process both of which are *not* dealt with in $\mathcal{KBDD}$.

Inference Algorithms for FDLs and Reasoning with GCIs. So far, reasoning in Fuzzy DLs has been mostly based on tableau-methods (e.g., [21, 19, 11, 20]). Most of these methods do not support reasoning with general terminologies as it is possible with $\mathbf{FixIt}(\mathbb{ALC})$. The first method ever to integrate GCIs into FDL reasoning is [19]. A very similar approach is presented in [11] for the fuzzy variant of a more expressive DL, namely $\mathcal{SHI}$. Very recently, [24] proposed a novel and elegant method for reasoning with GCIs (under a more general semantics than here) which is inspired by earlier works on tableau-based reasoning in multi-valued logics. The method combines a tableau-construction procedure with a Mixed Integer Linear Programming (MILP) solver that serves as an oracle to the FDL tableau procedure: tableau-expansion rules are used to create MILP problems which are sent for solution to the MILP solver. A solution to the MILP problems extends the structural information represented in the constructed completion forest by assignment of possibility degrees from which a model for the input KB can be derived. To the best of our knowledge there is no other approach to deal with GCIs in FDLs available at present. $\mathbf{FixIt}(\mathbb{ALC})$ therefore represents an interesting enrichment of inference calculi toolbox for FDLs, since no non-determinism is introduced by considering GCIs. A similar effect is achieved in [24] by the substantial modification of a standard tableau-based method and an extension with an MILP oracle: the tableau-expansion process does not become non-deterministic by introducing GCIs. However, depending on the solution techniques applied inside the MILP solver, non-determinism might simply be shifted from the tableau-construction process into the MILP oracle. In such cases, respective computational inefficiencies would then simply be hidden in then MILP oracle, but not actually resolved. A very similar approach that is not fixed to a specific semantics is presented in [7].

Further, [22] demonstrates how to use inference procedures for *classical* DLs to perform reasoning in (some) FDLs. This allows to use algorithms that have been developed for classical DLs in FDL reasoning (for some FDLs) in an indirect way. Please note that the $\mathcal{KBDD}$ procedure can not be used in such an indirect way to perform $\mathbb{ALC}$ reasoning, since both TBoxes and ABoxes are not supported.

[8, 10] consider a fuzzy version of $\mathcal{ALC}$ using arbitrary continuous t-norms (and the corresponding residuated implications) to define the semantics of the concept constructors and proposes a method for deciding $\emptyset \models \langle C \sqsubseteq D \geq 1\rangle$ and the satisfiability of $\langle C \sqsubseteq D \geq 1\rangle$ by mapping to a (decidable) propositional fuzzy logic. The generated propositional problems can be exponentially bigger than the FDL input problem. Although, the semantics considered in [8, 10] is more general than here (but differs for universal role restrictions), the proposed decision procedures cover more limited reasoning tasks, i.e. no background knowledge $\mathcal{K}$ is considered.

Complexity of FDL Reasoning. In comparison to classical DLs, remarkably little research has been done on the complexity of reasoning in FDLs. [21] considers $\mathbb{ALC}$ in a more restricted version than in this paper (i.e. the TBox in KBs are syntactically restricted) and shows that the entailment problem for fuzzy ABox assertions in this restricted case is PSPACE-complete. [9] studies the computational complexity of a class of fuzzy propositional logics (based on a class of continuous t-norms) and derives decidability (but not complexity) results for some validity and satisfiability related reasoning tasks in corresponding FDLs.

7 Conclusions and Future Work

We presented a novel procedure **FixIt**($\mathbb{ALC}$) for deciding knowledge base (KB) satisfiability in the FDL $\mathbb{ALC}$, introducing a *new class* of inference procedures into FDL reasoning. Besides the tableau-based methods [19, 11, 24, 7], it is the only (and the first non-tableau-based) approach to integrate general terminologies in FDL reasoning that we are aware of.

Additionally, we clarified the worst-case complexity of the reasoning problem that is solved by the algorithm and showed that deciding the satisfiability of a KB in $\mathbb{ALC}$ is an EXPTIME-complete problem. A discussion of a (straigthforward) implementation of the algorithm based on explicit representations of types shows that our algorithm can be implemented in a way that is worst-case optimal wrt. its runtime.

The main research questions that we want to address next are as follows: we will study means of implicit representation of sets of fuzzy types known from Symbolic Model Checking [12], in particular their implementation by means of Ordered Binary Decision Diagrams (OBDDs) [5] similar to [14], therefore addressing the main obstacle to apply the procedure in practice. A major question concerning optimization is clearly how to implement the final test of the algorithm efficiently, e.g. by heuristic search using the information in the ABox effectively to find the required mapping. The integration of optimizations such as full vs. lean representations or particle vs. types as discussed in [14] should be straightforward. We want to evaluate the effectiveness of the method by an implementation and comparison to tableau-based systems for FDLs. Moreover, we believe that it is interesting to study a bottom-up variant of $\mathcal{KBDD}$ in the context of FDLs too, and to check if the integration of ABoxes can be done more efficiently in such a variant. Finally, we would like to see to what extend the method can cover other semantics for FDLs (e.g. other t-norms) and extended constructs, such as fuzzy modifiers and concrete domains.

Acknowledgments

This work has been partially funded by the European Commission under the LarKC project (FP7 - 215535). Stijn Heymans is supported by the Austrian Science Fund (FWF) under projects P20305 and P20840.

References

[1] F. Baader, J. Hladik, and R. Peñaloza. SI! automata can show PSPACE results for description logics. In C. Martin-Vide, editor, *Proceedings of the First International Conference on Language and Automata Theory and Applications (LATA'07)*, Lecture Notes in Computer Science, 2007. To appear.

[2] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[3] Franz Baader, Ian Horrocks, and Ulrike Sattler. Description Logics. In Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors, *Handbook of Knowledge Representation*. Elsevier, 2007. To appear.

[4] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science (No. 53). Cambridge University Press, 2003.

[5] Randal E. Bryant. Symbolic Boolean manipulation with ordered binary-decision diagrams. *ACM Comput. Surv.*, 24(3):293–318, 1992.

[6] Francesco M. Donini and Fabio Massacci. EXPTIME tableaux for ALC. *Artif. Intell.*, 124(1):87–138, 2000.

[7] Volker Haarslev, Hsueh-Ieng Pai, and Nematollaah Shiri. Uncertainty Reasoning for Ontologies with General TBoxes in Description Logic. In Paulo C. G. Costa, Claudia D'Amato, Nicola Fanizzi, Kathryn B. Laskey, Ken Laskey, Thomas Lukasiewicz, Matthias Nickles, and Michael Pool, editors, *Uncertainty Reasoning for the Semantic Web I*, LNAI. Springer, 2008.

[8] Petr Hájek. Making fuzzy description logic more general. *Fuzzy Sets and Systems*, 154(1):1–15, 2005.

[9] Petr Hájek. Computational complexity of t-norm based propositional fuzzy logics with rational truth constants. *Fuzzy Sets and Systems*, 157(5):677–682, 2006.

[10] Petr Hájek. *What does Mathematical Fuzzy Logic offer to Description Logic?*, chapter in Fuzzy Logic and the Semantic Web. Capturing Intelligence. Elsevier, 2006.

[11] Yanhui Li, Baowen Xu, Jianjiang Lu, and Dazhou Kang. Discrete Tableau Algorithms for $\mathcal{FSHI}$. In *Proceedings of the International Workshop on Description Logics (DL)*, 2006.

[12] Kenneth L. McMillan. *Symbolic Model Checking*. Kluwer Academic Publishers, Norwell, MA, USA, 1993.

[13] Carlo Meghini, Fabrizio Sebastiani, and Umberto Straccia. A model of multimedia information retrieval. *Journal of the ACM*, 48(5):909–970, 2001.

[14] Guoqiang Pan, Ulrike Sattler, and Moshe Y. Vardi. BDD-based decision procedures for the modal logic K. *Journal of Applied Non-Classical Logics*, 16(1-2):169–208, 2006.

[15] Peter F. Patel-Schneider, Patrick Hayes, and Ian Horrocks. OWL Web Ontology Language Semantics and Abstract Syntax. Candidate Recommendation 18 August 2003, W3C, 2003.

[16] Vaughan R. Pratt. A Near-Optimal Method for Reasoning about Action. *J. Comput. Syst. Sci.*, 20(2):231–254, 1980.

[17] Klaus Schild. A correspondence theory for terminological logics: Preliminary report. In *In Proceedings of the International Joint Conference of Artificial Intelligence (IJCAI 1991)*, pages 466–471, 1991.

[18] Manfred Schmidt-Schauß and Gert Smolka. Attributive Concept Descriptions with Complements. *Artif. Intell.*, 48(1):1–26, 1991.

[19] George Stoilos, Umberto Straccia, George Stamou, and Jeff Pan. General Concept Inclusions in Fuzzy Description Logics. In *Proceedings of the 17th Eureopean Conference on Artificial Intelligence (ECAI-06)*, pages 457–461. IOS Press, 2006.

[20] Giorgos Stoilos, Giorgos B. Stamou, Jeff Z. Pan, Vassilis Tzouvaras, and Ian Horrocks. Reasoning with very expressive fuzzy description logics. *J. Artif. Intell. Res. (JAIR)*, 30:273–320, 2007.

[21] Umberto Straccia. Reasoning within Fuzzy Description Logics. *Journal of Artificial Intelligence Research*, 14:137–166, 2001.

[22] Umberto Straccia. Transforming Fuzzy Description Logics into Classical Description Logics. In *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA-04)*, number 3229 in Lecture Notes in Computer Science, pages 385–399, Lisbon, Portugal, 2004. Springer Verlag.

[23] Umberto Straccia. A Fuzzy Description Logic for the Semantic Web. In Elie Sanchez, editor, *Fuzzy Logic and the Semantic Web*, Capturing Intelligence, chapter 4, pages 73–90. Elsevier, 2006.

[24] Umberto Straccia and Fernando Bobillo. Mixed integer programming, general concept inclusions and fuzzy description logics. *Mathware & Soft Computing*, 14(3):247–259, 2007.

[25] Stephan Tobies. *Complexity results and practical algorithms for logics in Knowledge Representation*. Phd thesis, LuFG Theoretical Computer Science, RWTH-Aachen, Germany, 2001.

A Appendix: Change Log

August 7, 2008. First draft of the technical report.

August 10, 2008. Added Section 3.1 which formally proofs EXPTIME-completeness of the reasoning task which is solved by our proposed algorithm. Restructured the report slightly. Added a section on the classical DL $\mathcal{ALC}$.

August 19, 2008. Small improvements and fixes in Section 3.1. Changed the title of the report.

August 25, 2008. Added Section 4 discussing the integration of GCIs in standard tableau-based approaches for FDL reasoning and highlighting why the novel method proposed here might be an interesting alternative to tableau-based methods.

August 26, 2008. Extended Section 3.1 to Section 3.

September 2, 2008. Added Section 6 on related work. Updated discussion in Section 5.4 slighly.

September 8, 2008. Small fixes and improvements throughout the paper.

September 9, 2008. Completed Section 3.2 and added most of Section 3.3.

September 10, 2008. Added an outline of the document in section 1.

September 11, 2008. Extended the related work and references.

October 16, 2008. Integrated comments from reviewers of the Uncertainty Reasoning for the Semantic Web (URSW) 2008 workshop and the Logical Foundations of Computer Science (LFCS) 2009 conference.