Reasoning in Fuzzy Description Logics using Automata

Stefan Borgwardt[a,*], Rafael Peñaloza[b,*]

[a] *Institute for Theoretical Computer Science, Technische Universität Dresden, Germany*
[b] *KRDB Research Centre, Free University of Bozen-Bolzano, Italy*

Abstract

Automata-based methods have been successfully employed to prove tight complexity bounds for reasoning in many classical logics, and in particular in Description Logics (DLs). Very recently, the ideas behind these automata-based approaches were adapted for reasoning also in fuzzy extensions of DLs, with semantics based either on finitely many truth degrees or the Gödel t-norm over the interval $[0, 1]$. Clearly, due to the different semantics in these logics, the construction of the automata for fuzzy DLs is more involved than for the classical case.

In this paper we provide an overview of the existing automata-based methods for reasoning in fuzzy DLs, with a special emphasis on explaining the ideas and the requirements behind them. The methods vary from deciding emptiness of automata on infinite trees to inclusions between automata on finite words. Overall, we provide a comprehensive perspective on the automata-based methods currently in use, and the many complexity results obtained through them.

Keywords: Fuzzy Description Logics, Reasoning, Tree Automata, Weighted Automata

1. Introduction

The Web Ontology Language OWL 2[1] has been widely used to formulate many different ontologies, with a particular concern in the biomedical sciences and healthcare systems, as evidenced by the large ontology repository BioPortal.[2] This is due in no small part to the well-defined formal semantics of OWL 2 DL and its tractable profiles, which are provided by Description Logics (DLs) [1] of different expressivity, and the existence of many highly optimized reasoning systems for DLs.[3] OWL 2 DL and its profiles provide the flexibility for handling many different ontology development needs, and existing ontologies cover the whole range, from very large ontologies written over a very inexpressive language, like SNOMED CT,[4] to more specialized ones describing complex interactions between concepts.

In description logics, knowledge is represented using *concepts* describing sets of objects, such as Fever and Male, and *roles* that draw connections between objects, such as hasSymptom and hasParent. Formally, concepts correspond to unary predicates from first-order logic, while roles are binary predicates. In DLs, complex concepts are usually built from two disjoint sets of *concept names* and *role names* using *concept constructors*. The specific DL used, and hence its expressivity, is determined by the choice of constructors allowed.

The ontology itself expresses the domain knowledge through a finite set of axioms that restrict the way in which concepts may be interpreted, and state some explicit knowledge about the individuals populating

*Corresponding authors. Tel.: +49 351 463 38231; Fax: +49 351 463 37959.
Email addresses: stefborg@tcs.inf.tu-dresden.de (Stefan Borgwardt), rafael.penaloza@unibz.it (Rafael Peñaloza)

[1] http://www.w3.org/TR/owl2-overview/
[2] http://bioportal.bioontology.org/
[3] http://www.w3.org/2001/sw/wiki/OWL/Implementations
[4] http://www.ihtsdo.org/snomed-ct/

the domain. For example, the *general concept inclusion*

$$\forall \mathsf{hasParent}.\exists \mathsf{hasBloodType}.\mathsf{TypeO} \sqsubseteq \exists \mathsf{hasBloodType}.\mathsf{TypeO},$$

expresses that people whose parents both have blood type O must also have the same blood type. Similarly, the *assertions*

$$\mathsf{hasBloodType}(\mathtt{henry}, \mathtt{t}) \quad \text{and} \quad \mathsf{TypeA}(\mathtt{t})$$

state that the individual Henry has a blood type that can be classified as type A. Along with this explicitly represented knowledge, many implicit consequences can be derived from a given ontology. One of the main goals in DLs is to devise and implement efficient reasoning methods that can extract this implicit knowledge in an automated manner.

Since they attempt to model the real world, in biomedical ontologies it is common to find concepts such as $\mathsf{HighConcentration}$ (GALEN[5]) and $\mathsf{LowFrequency}$ (SNOMED CT) for which no precise definition can be given. For instance, there is no precise point at which a concentration stops being *normal* to become *high*. Fuzzy DLs have been introduced as knowledge representation formalisms capable of handling imprecise concepts, as well as imprecise knowledge. To achieve this, they introduce more than two degrees of truth, allowing for a more fine-grained analysis of interactions between such vague concepts. For example, given the knowledge

$$\exists \mathsf{hasDisease}.\mathsf{Influenza} \sqsubseteq \exists \mathsf{hasSymptom}.\mathsf{Fever} \sqcap \exists \mathsf{hasSymptom}.\mathsf{Headache}$$

that influenza induces the symptoms fever and headache, it makes sense to say that a lower degree of fever indicates a less severe case of influenza, which is modeled by using a lower degree of truth.

Fuzzy DLs are based on fuzzy set theory [2] and mathematical fuzzy logic [3]. Usually, the set of truth degrees is the interval $[0, 1]$ of real numbers, and thus a statement can be true to some intermediate degree, e.g. 0.5, instead of only completely true (1) or completely false (0). The logical constructors such as conjunction and implication are then generalized to this extended set of truth degrees. The oldest and most popular fuzzy semantics, the so-called *Zadeh* semantics, is based on the minimum function to interpret conjunction, the maximum function for disjunction, $1 - x$ for the negation $\neg x$, and the Kleene-Dienes-implication $\max\{1 - x, y\}$ for the implication $x \to y$. The latter is derived from the classical equivalence of $x \to y$ and $\neg x \vee y$. However, this choice of implication function has some unintuitive consequences [4]. This is particularly unfortunate in the context of description logics since they make heavy use of implications in the axioms ($\sqsubseteq$) and some constructors.

An alternative approach with a deeper formal motivation uses arbitrary *t-norms*, which satisfy only some basic properties, as interpretation functions for the conjunction [5]. The implication is then evaluated using the *residuum* of such a t-norm. The minimum function is a t-norm, also called the *Gödel t-norm*, but there are infinitely many others. Unfortunately, using t-norm based semantics for fuzzy DLs turned out to be problematic, with very inexpressive DLs losing their tractability [6], and slightly more expressive logics becoming undecidable [7–10]. Two possible ways to avoid this problem that have been identified are to restrict the semantics either to the (relatively) harmless Gödel t-norm or to a finite set of truth degrees.

Automata-based methods have proven to be very useful tools for finding tight bounds for the complexity of reasoning in DLs and other logics [11–13]. In a nutshell, automata provide an elegant and complexity-wise efficient formalism for handling complex, potentially infinite, models. Very recently, the ideas developed for the classical case have been adapted for proving tight complexity bounds for reasoning in DLs using the Gödel t-norm [14, 15], or a finite structure [16] for their semantics. Perhaps surprisingly, reasoning in these fuzzy logics does not require the use of fuzzy automata [17], but standard non-deterministic automata suffice. However, the translation to these automata is necessarily more elaborate to be able to handle models with infinitely many different truth degrees (as in [14]) and complex finitely-valued t-norms (see [16]).

In this paper, we present an overview of the automata-based methods recently developed for reasoning in these fuzzy DLs. In Section 4.1, we describe how to find tree-shaped fuzzy models directly through the runs of an automaton, an approach which was developed in [16, 18]. Section 4.2 illustrates a case where

[5] `http://www.opengalen.org/`

this model cannot be constructed directly, but instead an abstraction of such a model can be recognized by an automaton; see [14]. Finally, Section 5 is concerned with a different approach, where the interpretations of concepts can be characterized in terms of regular languages recognized by finite word automata [15]. Our goal is to provide a comprehensive perspective of the ideas employed and the results obtained. To simplify the description and improve readability, we consider only relatively inexpressive fuzzy DLs which, nonetheless, are expressive enough to motivate the different constructions. Due to our general definitions of fuzzy ontologies and reasoning problems, we provide new (although not surprising) results for reasoning w.r.t. *ordered* ABoxes in finitely-valued DLs (Theorems 16, 28, and 29). In Section 6, we describe how far these approaches can be generalized to more expressive logics and other types of ontologies. The interested reader can find all the proofs, in a more general context, in the original papers [14–16].

2. Gödel Fuzzy Description Logics

Following the approach from Mathematical Fuzzy Logic [5], fuzzy Description Logics (DLs) extend classical DLs by allowing more than two truth (or membership) *degrees*. For the sake of clarity, here we consider the most prominent approaches, in which the class of truth degrees is either the whole interval of real numbers $[0, 1]$, or a finite subset $L \subseteq [0, 1]$ of this interval that includes the extreme values 0 and 1. However, readers should be aware that other classes of truth degrees have also been considered in the literature; see Section 6 for more details.

In order to handle more than two truth degrees, the interpretation of the logical connectives is extended in a truth-functional manner. In general there exist infinitely many possible meaningful interpretations, which have been extensively studied in the literature [3, 5, 19]. One of the best known semantics, due to its simplicity and good behavior, is the so-called *Gödel* semantics, in which conjunction is interpreted by the minimum of the truth degrees, and the interpretations of all other connectives are guaranteed to maintain basic logical properties in relation to this function.

We now formally introduce the syntax and semantics of fuzzy DLs under the Gödel semantics. Since the family of Description Logics has many members, we consider here only fuzzy extensions of the basic DL $\mathcal{ALC}$ and its sublogic $\mathcal{FL}_0$ as prototypical examples. In Section 6 we discuss which of the presented results can be extended to other (more expressive) DLs. The semantics of the logics depends on the class of truth degrees available. Thus, for the rest of this paper we consider a fixed, but arbitrary, chain $\mathfrak{C}$ of truth degrees, which is either the interval $[0, 1]$, or a *finite* subset $L \subseteq [0, 1]$ including 0 and 1. In the general case, we denote the resulting fuzzy DLs using the prefix $\mathfrak{C}$- (e.g. $\mathfrak{C}$-$\mathcal{FL}_0$), for the interval $[0, 1]$ we adopt the notation G- (for *Gödel*), and when a finite set L is considered, we denote it by L-.

2.1. Syntax and Semantics of $\mathfrak{C}$-$\mathfrak{N}\mathcal{AL}$

As in all DLs, the syntax of the fuzzy description logic $\mathfrak{C}$-$\mathfrak{N}\mathcal{AL}$ is based on concepts and roles.[6] These are interpreted as (fuzzy) unary and binary relations, respectively. More formally, let $\mathsf{N_I}$, $\mathsf{N_R}$, and $\mathsf{N_C}$ be mutually disjoint sets of *individual*, *role*, and *concept names*, respectively. $\mathfrak{C}$-$\mathfrak{N}\mathcal{AL}$ *concepts* are built through the rule

$$C ::= A \mid \top \mid \neg C \mid C \sqcap C \mid \exists r.C \mid \forall r.C,$$

where $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$. We call concepts of the form $\exists r.C$ or $\forall r.C$ *quantified concepts*. The semantics of this logic is given by means of interpretations, which are defined next.

Definition 1 (interpretation)**.** An *interpretation* is a pair $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty *domain*, and $\cdot^\mathcal{I}$ is a function that maps every individual name $a \in \mathsf{N_I}$ to an element $a^\mathcal{I} \in \Delta^\mathcal{I}$, every concept name $A \in \mathsf{N_C}$ to a fuzzy set $A^\mathcal{I} \colon \Delta^\mathcal{I} \to \mathfrak{C}$, and every role name $r \in \mathsf{N_R}$ to a fuzzy binary relation $r^\mathcal{I} \colon \Delta^\mathcal{I} \times \Delta^\mathcal{I} \to \mathfrak{C}$.

[6] $\mathfrak{C}$-$\mathfrak{N}\mathcal{AL}$ is a fuzzy extension of the classical DL $\mathcal{ALC}$. The difference in the name is chosen to emphasize that the residual negation, as opposed to the involutive negation $x \mapsto 1-x$, is used for its semantics, and that there is no disjunction constructor.

Table 1: Syntax and Semantics of $\mathfrak{C}$-$\mathfrak{N}\mathcal{AL}$

Constructor	Syntax	Semantics
top concept	$\top$	1
(residual) negation	$\neg C$	$C^{\mathcal{I}}(x) \Rightarrow 0$
conjunction	$C \sqcap D$	$\min(C^{\mathcal{I}}(x), D^{\mathcal{I}}(x))$
existential restriction	$\exists r.C$	$\sup_{y\in\Delta^{\mathcal{I}}} \min(r^{\mathcal{I}}(x,y), C^{\mathcal{I}}(y))$
value restriction	$\forall r.C$	$\inf_{y\in\Delta^{\mathcal{I}}} r^{\mathcal{I}}(x,y) \Rightarrow C^{\mathcal{I}}(y)$

This function is extended to arbitrary concepts using the Gödel operators as shown in Table 1, where $\Rightarrow$ is the *Gödel residuum* defined for every $p, q \in \mathfrak{C}$ as

$$p \Rightarrow q := \begin{cases} q & \text{if } p > q \\ 1 & \text{otherwise.} \end{cases}$$

The residuum has the interesting property that for all $x, y, z \in \mathfrak{C}$ we have $\min(x, y) \leq z$ iff $y \leq x \Rightarrow z$, which is shared by all continuous t-norms and their residua [19]. The *residual negation* $x \Rightarrow 0$ is always crisp, i.e. evaluated to 0 or 1:

$$x \Rightarrow 0 = \begin{cases} 0 & \text{if } x > 0 \\ 1 & \text{otherwise.} \end{cases}$$

Notice that the semantics of the quantified concepts requires a computation over the whole (potentially infinite) domain. To handle this computation effectively, in the literature on fuzzy DLs interpretations are usually restricted to be witnessed [20], which means that existential and value restrictions must be interpreted as maxima and minima, respectively. More formally, an interpretation $\mathcal{I}$ is *witnessed* if for every existential restriction $\exists r.C$ and every $x \in \Delta^{\mathcal{I}}$ there is a *witness* $y \in \Delta^{\mathcal{I}}$ such that $(\exists r.C)^{\mathcal{I}}(x) = \min(r^{\mathcal{I}}(x, y), C^{\mathcal{I}}(y))$, and analogously for every value restriction $\forall r.C$ and every $x \in \Delta^{\mathcal{I}}$ there is a witness $y \in \Delta^{\mathcal{I}}$ with $(\forall r.C)^{\mathcal{I}}(x) = (r^{\mathcal{I}}(x, y) \Rightarrow C^{\mathcal{I}}(y))$. We also adopt this restriction here, and for the rest of this paper consider only witnessed interpretations. Notice that considering witnessed interpretations is only a restriction in the case that the class of truth degrees is infinite. Whenever only finitely many truth degrees are used, every interpretation is necessarily witnessed [20]. For brevity, we will henceforth call witnessed interpretations simply *interpretations*.

The main use of description logics is to serve as languages for knowledge representation. The knowledge of a domain is represented using axioms that restrict the class of interpretations that are relevant for the different reasoning tasks. These axioms may describe general relations between concepts, or properties of some specific individuals.

Definition 2 (axioms)**.** A *crisp assertion* is either a *concept assertion* of the form $a{:}C$ or a *role assertion* of the form $(a, b){:}r$ for a concept C, $r \in \mathsf{N_R}$, and $a, b \in \mathsf{N_I}$. An *(order) assertion* is an expression of the form $\langle \alpha \bowtie \beta \rangle$, where α is a crisp assertion, β is either a crisp assertion or a value from $\mathfrak{C}$, and $\bowtie \in \{=, \geq, >, \leq, <\}$. The interpretation $\mathcal{I}$ *satisfies* the order assertion $\langle \alpha \bowtie \beta \rangle$ iff $\alpha^{\mathcal{I}} \bowtie \beta^{\mathcal{I}}$, where $(a{:}C)^{\mathcal{I}} := C^{\mathcal{I}}(a^{\mathcal{I}})$, $((a, b){:}r)^{\mathcal{I}} := r^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}})$, and $p^{\mathcal{I}} := p$ for all $p \in \mathfrak{C}$. An *ordered ABox* $\mathcal{A}$ is a finite set of order assertions. An interpretation is a *model* of the ordered ABox $\mathcal{A}$ if it satisfies all order assertions in $\mathcal{A}$.

A *general concept inclusion (GCI)* is an expression of the form $\langle C \sqsubseteq D \geq p \rangle$, where C and D are concepts and $p \in \mathfrak{C}$. The interpretation $\mathcal{I}$ satisfies this GCI if for all $x \in \Delta^{\mathcal{I}}$ it holds that $C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x) \geq p$. A *TBox* is a finite set of GCIs. An *ontology* is a pair $\mathcal{O} = (\mathcal{A}, \mathcal{T})$, where $\mathcal{A}$ is an ordered ABox and $\mathcal{T}$ is a TBox. An interpretation is a *model* of a TBox $\mathcal{T}$ if it satisfies all GCIs in $\mathcal{T}$, and it is a *model* of an ontology $\mathcal{O} = (\mathcal{A}, \mathcal{T})$ if it is a model of both $\mathcal{A}$ and $\mathcal{T}$.

Notice that the condition for the interpretation $\mathcal{I}$ to satisfy the GCI $\langle C \sqsubseteq D \geq p \rangle$ is equivalent to requiring, for every $x \in \Delta^{\mathcal{I}}$ that either $C^{\mathcal{I}}(x) \leq D^{\mathcal{I}}(x)$ or $D^{\mathcal{I}}(x) \geq p$. In particular, if $p = 1$, then this restriction simplifies to requiring that $C^{\mathcal{I}}(x) \leq D^{\mathcal{I}}(x)$. For brevity, we will usually write GCIs of the form $\langle C \sqsubseteq D \geq 1\rangle$ as $\langle C \sqsubseteq D\rangle$.

In the literature, often only assertions of the form $\langle \alpha \bowtie p\rangle$ with $p \in \mathfrak{C}$ are allowed (see e.g. [21]).[7] We consider a more expressive variant that can express order relations between crisp assertions. For example, we can express that an individual a belongs to the concept C with a higher truth degree than the individual b through the assertion $\langle a : C > b : C\rangle$ without having to specify any explicit bounds on the degrees of $C^{\mathcal{I}}(a^{\mathcal{I}})$ or $C^{\mathcal{I}}(b^{\mathcal{I}})$. This more general kind of ABox will also be helpful for developing a simpler reasoning algorithm for this logic, as shown in Section 4.

We use the expression $\mathsf{sub}(\mathcal{O})$ to denote the set of all subconcepts appearing in an ontology $\mathcal{O}$, either as part of a GCI or in a concept assertion. Clearly, since $\mathcal{O}$ is a finite set of axioms, $\mathsf{sub}(\mathcal{O})$ is always finite. We further denote by $\mathcal{V}_{\mathcal{O}}$ the set of all truth degrees appearing in $\mathcal{O}$, together with 0 and 1. This set is also always finite and totally ordered. We often denote the elements of $\mathcal{V}_{\mathcal{O}} \subseteq \mathfrak{C}$ as $0 = p_0 < p_1 < \cdots < p_k = 1$, where $|\mathcal{V}_{\mathcal{O}}| = k + 1$.

The most basic reasoning task in $\mathfrak{C}$-$\mathfrak{N}\mathcal{AL}$ and other (fuzzy) DLs is to decide consistency of a set of axioms; that is, whether a given ontology has a (witnessed) model. However, several other decision problems and inferences exist. Moreover, one might also be interested in computing the truth degree of an inference. All these tasks are formalized next.

Definition 3 (reasoning)**.** An ontology $\mathcal{O}$ is *consistent* if it has a model. Given $p \in \mathfrak{C}$, a concept C is *p-satisfiable* w.r.t. $\mathcal{O}$ if there is a model $\mathcal{I}$ of $\mathcal{O}$ and an $x \in \Delta^{\mathcal{I}}$ with $C^{\mathcal{I}}(x) \geq p$. The *best satisfiability degree* of C w.r.t. $\mathcal{O}$ is the supremum over all p such that C is p-satisfiable w.r.t. $\mathcal{O}$.

C is *p-subsumed* by a concept D w.r.t. $\mathcal{O}$ if all models of $\mathcal{O}$ satisfy the GCI $\langle C \sqsubseteq D \geq p\rangle$. The *best subsumption degree* of C and D w.r.t. $\mathcal{O}$ is the supremum over all p such that C is p-subsumed by D w.r.t. $\mathcal{O}$.

If consistency is decidable, then satisfiability and subsumption can be restricted without loss of generality to ontologies containing an empty ABox: if $\mathcal{O}$ is inconsistent, then these two problems are trivial since $\mathcal{O}$ has no model, and if $\mathcal{O}$ is consistent, then the ABox assertions cannot contradict the p-satisfiability of C, and therefore C is p-satisfiable w.r.t. $\mathcal{O} = (\mathcal{A}, \mathcal{T})$ iff it is p-satisfiable w.r.t. $(\emptyset, \mathcal{T})$. A similar argument can be made for subsumption-related reasoning tasks. For details, see [24, Lemma 2.22], where this claim is shown for ordinary ABoxes; however, the proof can easily be adapted to ordered ABoxes.

2.2. $\mathfrak{C}$-$\mathcal{FL}_0$ and Greatest Fixed-Point Semantics

The second class of logics we consider in this paper is denoted by $\mathfrak{C}$-$\mathcal{FL}_0$ and obtained from $\mathfrak{C}$-$\mathfrak{N}\mathcal{AL}$ by allowing only top, conjunctions, and value restrictions as concept constructors. More precisely, $\mathfrak{C}$-$\mathcal{FL}_0$ concepts are built using the rule

$$C ::= A \mid \top \mid C \sqcap C \mid \forall r.C.$$

The notions of axioms, ontologies, and their semantics are defined analogously to $\mathfrak{C}$-$\mathfrak{N}\mathcal{AL}$ over this restricted class of concepts. Since this logic is incapable of expressing negations, consistency can be decided by a non-deterministic algorithm in a straightforward manner: simply guess an ordering of all the assertions appearing in the ABox, and verify that it complies with the restrictions in the TBox. Thus, when dealing with $\mathfrak{C}$-$\mathcal{FL}_0$, we focus on deciding subsumption and computing the best subsumption degree w.r.t. a TBox. Adapting from [11], we further restrict the TBox to contain only a special class of GCIs called *primitive concept definitions* and consider greatest fixed-point semantics, which leads to a lower complexity for the reasoning problems.

Definition 4 (cyclic TBox)**.** A *(primitive concept) definition* is a GCI of the form $\langle A \sqsubseteq C \geq p\rangle$, where $A \in \mathsf{N_C}$, C is a concept, and $p \in \mathfrak{C}$. A *cyclic TBox* is a finite set of definitions. Given a cyclic TBox $\mathcal{T}$, a concept name is *defined* if it appears on the left-hand side of some definition in $\mathcal{T}$, and *primitive* otherwise.

[7] Notable exception are the *fuzzyDL* reasoner, which allows to state arithmetic constraints over individuals [22], and the formalism of [23], which allows to use rather expressive comparison expressions as (crisp) concepts.

Note that, as usual in the DL literature [1], cyclic TBoxes do not necessarily contain a cycle in the concept definitions; however, cycles are allowed. It is easy to see that every cyclic TBox can be equivalently rewritten into one where all the definitions are of the form $\langle A \sqsubseteq \forall w.B \geq p\rangle$, where $A, B \in \mathsf{N_C}$, $w \in \mathsf{N_R^*}$, and $p \in \mathfrak{C}$,[8] and there are no two definitions $\langle A \sqsubseteq \forall w.B \geq p\rangle$, $\langle A \sqsubseteq \forall w.B \geq p'\rangle$ with $p \neq p'$ [15]. We say that such a TBox is in *normal form*. Concept definitions can be seen as a restriction of the interpretation of the defined concepts, depending on the interpretation of the primitive concepts. We use this intuition and consider *greatest fixed-point* semantics. The following construction is based on the classical notions from [11].

Let now Δ be a fixed domain. A *primitive interpretation* is a pair $\mathcal{J} = (\Delta, \cdot^{\mathcal{J}})$ as in Definition 1, except that the interpretation function $\cdot^{\mathcal{J}}$ is only defined on $\mathsf{N_R}$ and the set $\mathsf{N_P^{\mathcal{T}}}$ of all primitive concept names w.r.t. $\mathcal{T}$. Given such a primitive interpretation $\mathcal{J}$, we use functions $f\colon \mathsf{N_D^{\mathcal{T}}} \to \mathfrak{C}^{\Delta}$ to describe possible interpretations for the set of defined concept names $\mathsf{N_D^{\mathcal{T}}}$. Given a primitive interpretation $\mathcal{J}$ and a function $f\colon \mathsf{N_D^{\mathcal{T}}} \to \mathfrak{C}^{\Delta}$, the *induced interpretation* $\mathcal{I}_{\mathcal{J},f}$ has the same domain as $\mathcal{J}$ and extends the interpretation function of $\mathcal{J}$ to the defined concepts names $A \in \mathsf{N_D^{\mathcal{T}}}$ by taking $A^{\mathcal{I}_{\mathcal{J},f}} := f(A)$. The interpretation of the remaining concept names, i.e., those that do not occur in $\mathcal{T}$, is fixed to map all elements of the domain to 0.

Let $L^{\mathcal{T}} := (\mathfrak{C}^{\Delta})^{\mathsf{N_D^{\mathcal{T}}}}$ denote the set of all functions f of the form described above. We can describe the effect that the axioms in $\mathcal{T}$ and the primitive interpretation $\mathcal{J}$ have on $L^{\mathcal{T}}$ through the operator $T_{\mathcal{J}}^{\mathcal{T}}\colon L^{\mathcal{T}} \to L^{\mathcal{T}}$, which is defined as follows for all $f \in L^{\mathcal{T}}$, $A \in \mathsf{N_D^{\mathcal{T}}}$, and $x \in \Delta$:

$$T(f)(A)(x) := \inf_{\langle A \sqsubseteq C \geq p\rangle \in \mathcal{T}} (p \Rightarrow C^{\mathcal{I}_{\mathcal{J},f}}(x)).$$

This operator computes new values of the defined concept names according to the old interpretation $\mathcal{I}_{\mathcal{J},f}$ and their definitions in $\mathcal{T}$.

We are interested in using the greatest fixed-point of $T_{\mathcal{J}}^{\mathcal{T}}$, for some primitive interpretation $\mathcal{J}$, to define a new semantics for cyclic TBoxes $\mathcal{T}$ in $\mathfrak{C}$-$\mathcal{FL}_0$. Obviously, this can only be achieved if such a fixed-point exists. As for the classical case, it can be shown that the operator $T_{\mathcal{J}}^{\mathcal{T}}$ is downward ω-continuous and hence, using a basic result from Tarski [25], $T_{\mathcal{J}}^{\mathcal{T}}$ has a greatest fixed-point which can be computed by an iterative application of $T_{\mathcal{J}}^{\mathcal{T}}$ to the function that maps all defined concept names to $\top^{\mathcal{J}}$ (see [11, 15] for details).

In the following, we denote by $\mathsf{gfp}_{\mathcal{T}}(\mathcal{J})$ the interpretation $\mathcal{I}_{\mathcal{J},f}$ for $f := \mathsf{gfp}(T_{\mathcal{J}}^{\mathcal{T}})$. Note that $\mathcal{I} := \mathsf{gfp}_{\mathcal{T}}(\mathcal{J})$ is actually a model of $\mathcal{T}$ since for every $\langle A \sqsubseteq C \geq p\rangle \in \mathcal{T}$ and every $x \in \Delta$ we have

$$A^{\mathcal{I}}(x) = f(A)(x) = T_{\mathcal{J}}^{\mathcal{T}}(f)(A)(x) = \inf_{\langle A \sqsubseteq C' \geq p'\rangle \in \mathcal{T}} (p' \Rightarrow C'^{\mathcal{I}}(x)) \leq p \Rightarrow C^{\mathcal{I}}(x)$$

since f is a fixed-point of $T_{\mathcal{J}}^{\mathcal{T}}$, and thus $\min(p, A^{\mathcal{I}}(x)) \leq C^{\mathcal{I}}(x)$, which is equivalent to $p \leq A^{\mathcal{I}}(x) \Rightarrow C^{\mathcal{I}}(x)$.

We can now define the reasoning problem in $\mathfrak{C}$-$\mathcal{FL}_0$ that we want to solve.

Definition 5 (gfp-subsumption)**.** An interpretation $\mathcal{I}$ is a *gfp-model* of a TBox $\mathcal{T}$ if there is a primitive interpretation $\mathcal{J}$ such that $\mathcal{I} = \mathsf{gfp}_{\mathcal{T}}(\mathcal{J})$. Given $A, B \in \mathsf{N_C}$ and $p \in \mathfrak{C}$, A is *gfp-subsumed* by B to degree p w.r.t. $\mathcal{T}$ if every gfp-model $\mathcal{I}$ of $\mathcal{T}$ satisfies the GCI $\langle A \sqsubseteq B \geq p\rangle$. The *best gfp-subsumption degree* of A and B w.r.t. $\mathcal{T}$ is the supremum over all p such that A is gfp-subsumed by B to degree p w.r.t. $\mathcal{T}$.

The normalization of cyclic TBoxes can be made such that equivalence w.r.t. gfp-semantics is preserved. Therefore the gfp-models of the cyclic TBox $\mathcal{T}$ are the same as those of some cyclic TBox $\mathcal{T}'$ in normal form whose size is linear in the size of $\mathcal{T}$. To solve the problem of deciding gfp-subsumptions, it thus suffices to consider TBoxes in normal form.

Recall that the class of truth degrees $\mathfrak{C}$ may be a finite subset $L \subseteq [0, 1]$ containing the extreme elements 0 and 1, or the whole interval of real numbers $[0, 1]$. Whenever it is important to distinguish the shape of $\mathfrak{C}$ used for the semantics of the logic, we will denote it explicitly in the name of the logic and write G-$\mathfrak{NAL}$ if $\mathfrak{C} = [0, 1]$, and L-$\mathfrak{NAL}$ if $\mathfrak{C}$ is the finite subset L.

We demonstrate later how automata can be used to solve reasoning tasks in the different variants of $\mathfrak{C}$-$\mathfrak{NAL}$. In the next section, we provide a basic introduction to the automata models that we use and their associated decision problems.

[8] For $w = r_1 \ldots r_n \in \mathsf{N_R^*}$, we use $\forall w.B$ to abbreviate $\forall r_1 . \ldots \forall r_n.B$ ($\forall \varepsilon.B$ is simply B).

3. Basic Automata Theory

We now describe the automata models that will be the base for our reasoning techniques. In general, one can think of finite-state automata as machines that recognize languages defined over different kinds of structures. For the scope of this paper, we focus on two automata models that work on finite words and infinite trees, respectively.

3.1. Automata on Finite Words

One of the best known classes of automata is that of finite (word) automata [26]. Intuitively, these are non-deterministic machines that read an input word only once, one symbol at a time, and accept or reject it depending on the state they reach. For the scope of this paper, we consider a variant of finite automata that can read several symbols simultaneously.

Definition 6 (WA)**.** An *automaton with word transitions (WA)* is a tuple $\mathbf{A} = (\Sigma, Q, q_0, \Delta, q_f)$, where

- Σ is a finite alphabet of *input symbols*,
- Q is a finite set of *states*
- $q_0 \in Q$ is the *initial state*,
- $\Delta \subseteq Q \times \Sigma^* \times Q$ is the finite *transition relation*, and
- $q_f \in Q$ is the *final state.*

A *finite path* in $\mathbf{A}$ is a sequence $\pi = q_0 w_1 q_1 w_2 \dots w_n q_n$, where $q_i \in Q$ and $w_i \in \Sigma^*$ for all $i \in \{1, \dots, n\}$, and $q_n = q_f$. Its *label* is the finite word $\ell(\pi) := w_1 w_2 \dots w_n$. The path π is *accepting* if $(q_{i-1}, w_i, q_i) \in \Delta$ for all $i, 1 \leq i \leq n$. The set of all accepting finite paths (or *runs*) with label w in $\mathbf{A}$ is denoted by $\mathsf{paths}(\mathbf{A}, w)$. The *language accepted* by $\mathbf{A}$ is defined as $\mathcal{L}(\mathbf{A}) := \{w \in \Sigma^* \mid \mathsf{paths}(\mathbf{A}, w) \neq \emptyset\}$.

The idea is that, starting from a fixed initial state q_0, the automaton reads substrings of the input word w and non-deterministically chooses a new state according to its transition relation Δ. If after reading the whole word the automaton is in the final state, then w is accepted by $\mathbf{A}$. Clearly, every WA can be translated into a finite automaton that accepts the same language, by simply adding more states for reading each substring one symbol at a time. However, WA provide a clearer a more compact representation when specific words are read at a time, as is the case in this paper.

Automata with word transitions were used in [11] to decide subsumption between two (classical) $\mathcal{FL}_0$ concepts w.r.t. a cyclic TBox using the greatest fixed-point semantics. This is achieved by reducing the subsumption problem to a polynomial number of language inclusion tests between automata with word transitions. Thus, the complexity of deciding subsumption in classical $\mathcal{FL}_0$ is bounded from above by the complexity of deciding inclusion between the languages accepted by two WA, which is in PSPACE.

Proposition 7 ([27])**.** *Let $\mathbf{A}$, $\mathbf{A}'$ be two WA. Deciding whether $\mathcal{L}(\mathbf{A}) \subseteq \mathcal{L}(\mathbf{A}')$ requires polynomial space in the sizes of $\mathbf{A}$ and $\mathbf{A}'$.*

To reason about the more expressive constructors of $\mathfrak{NAL}$, we need a more complex automata model. We extend WA in two ways: first, the input structure is allowed to branch in the shape of a tree, and second, the input, and hence also the runs of the automaton, are infinite.

3.2. Automata on Infinite Trees

Tree automata extend finite automata by allowing the input structure to be tree-shaped. In particular, this means that the language accepted by such an automaton is a set of trees. We further extend this notion to consider infinite inputs. However, we also remove the alphabet symbols for the input trees, which means that there is only one input structure—the unlabeled tree $\{1, \dots, k\}^*$ for a fixed arity k. In this representation, ε is considered the *root node* of the tree, and ui with $u \in \{1, \dots, k\}^*$ and $i \in \{1, \dots, k\}$ is the *i-th successor* of u.

Definition 8 (LA)**.** A *looping automaton on k-ary infinite trees*, or *looping automaton* (LA) for short, is a tuple $\mathbf{A} = (Q, I, \Delta)$ consisting of

- a non-empty set Q of *states*,
- a subset $I \subseteq Q$ of *initial states*, and
- a *transition relation* $\Delta \subseteq Q^{k+1}$.

A *run* of this automaton is a mapping $\rho\colon \{1, \dots, k\}^* \to Q$ that satisfies the following two conditions:

1. $\rho(\varepsilon) \in I$, and
2. for all $u \in \{1, \dots, k\}^*$, it holds that $\big(\rho(u), \rho(u1), \dots, \rho(uk)\big) \in \Delta$.

We say that the LA $\mathbf{A}$ is *non-empty* iff it has a run.

Intuitively, a looping automaton tries to label an k-ary infinite tree in such a way that the root is labeled with an initial state (from I), and the labels of the successors of each node satisfy the transition relation Δ. If such a labeling is possible, then the automaton accepts the (unique) unlabeled input tree, and hence is non-empty; otherwise, it is empty.

Clearly, as the input tree is infinite, one cannot expect to produce the run (i.e., the labeling) explicitly in finite time. However, it is possible to decide whether $\mathbf{A}$ is non-empty using time bounded polynomially in the number of states of $\mathbf{A}$. The idea is simply to identify all *bad states*—those that cannot appear in any run—and then verify that there is at least one initial state that is not bad. For more details, see [12].

Proposition 9 ([12])**.** *Emptiness of an LA* $\mathbf{A}$ *can be decided in time polynomial in the size of* $\mathbf{A}$.

In the following section, we show how looping automata can be used to reason with $\mathfrak{C}$-$\mathfrak{N}\mathcal{AL}$ ontologies. Afterwards, we consider the easier case of greatest fixed-point semantics in $\mathfrak{C}$-$\mathcal{FL}_0$, which can be solved using automata with word transitions.

4. Reasoning with $\mathfrak{C}$-$\mathfrak{N}\mathcal{AL}$ Ontologies

We now consider the problem of reasoning with $\mathfrak{C}$-$\mathfrak{N}\mathcal{AL}$. As we will see, handling the infinite case will require a more elaborate technique than the finite case, and for that reason we study these cases separately. We proceed in this section as follows. We first tackle the problem of deciding consistency of an ontology. To achieve this, we introduce a simpler version of the problem that we call *local consistency*, in which the ABox is restricted to consider only one individual (see Definition 10). In Section 4.1 we use an LA to decide local consistency for L-$\mathfrak{N}\mathcal{AL}$ ontologies; i.e., when the set of truth degrees is finite. Section 4.2 shows that the same idea cannot be used directly for G-$\mathfrak{N}\mathcal{AL}$, and provides a new construction capable of handling the infinite set of truth degrees. Finally, in Section 4.3 we show how local consistency can be used to solve all the other reasoning tasks introduced in Section 2.

Definition 10 (local ABox)**.** A *local ABox* is an ordered ABox that contains no role assertions, and where all concept assertions refer to the same individual name. A *local ontology* is an ontology $(\mathcal{A}, \mathcal{T})$, where $\mathcal{A}$ is a local ABox. *Local consistency* refers to the problem of deciding consistency of a local ontology.

For example, the ABox $\{\langle \mathtt{henry}\colon \mathsf{Tall} > \mathtt{henry}\colon \mathsf{Fat}\rangle\}$ is local, while $\{\langle \mathtt{henry}\colon \mathsf{Tall} > \mathtt{tom}\colon \mathsf{Tall}\rangle\}$ and $\{\langle(\mathtt{henry}, \mathtt{henry})\colon \mathsf{likes}\rangle\}$ are not. As we see next, looping automata can be used for deciding local consistency in $\mathfrak{C}$-$\mathfrak{N}\mathcal{AL}$ by characterizing tree-shaped models of the local ontology as the runs of such an LA.

4.1. Local Consistency in L-$\mathfrak{NAL}$

Our algorithm for deciding local consistency is based on the fact that a local L-$\mathfrak{NAL}$ ontology $\mathcal{O}$ has a model iff it has a well-structured tree-shaped model, called a *Hintikka tree.* Intuitively, Hintikka trees are abstract representations of models that explicitly express the membership value of all the concepts that are relevant for the input ontology; i.e., those in $\mathsf{sub}(\mathcal{O})$. We construct automata that have exactly these Hintikka trees as their runs, and the initial states verify the existence of an element that satisfies the local ABox. Reasoning is hence reduced to testing emptiness of these looping automata.

Hintikka trees are infinite labeled trees, where each node is labeled with a so-called Hintikka function. These functions are fuzzy sets over the domain $\mathsf{sub}(\mathcal{O}) \cup \{\rho\}$, where ρ is an arbitrary new element that will be used to express the degree with which the role relation to the parent node holds. Hintikka functions restrict the values for complex concepts built using the propositional constructors $\top$, $\sqcap$, and $\neg$, according to the fuzzy semantics. The interpretation of existential and value restrictions is considered later, in Definition 12.

Definition 11 (Hintikka function)**.** A *Hintikka function* for $\mathcal{O}$ is a function $H\colon \mathsf{sub}(\mathcal{O}) \cup \{\rho\} \to L$ that satisfies the following conditions for every $C \in \mathsf{sub}(\mathcal{O})$:

1. $C = \top$ implies $H(C) = 1$,
2. if $C = D_1 \sqcap D_2$, then $H(C) = \min(H(D_1), H(D_2))$, and
3. if $C = \neg D$, then $H(C) = H(D) \Rightarrow 0$.

The Hintikka function H is *compatible* with the TBox $\mathcal{T}$ if for every GCI $\langle C \sqsubseteq D \geq p\rangle \in \mathcal{T}$ it holds that $H(C) \Rightarrow H(D) \geq p$. It is *compatible* with $\mathcal{A}$ if for every order assertion $\langle a{:}C \bowtie a{:}D\rangle$ or $\langle a{:}C \bowtie p\rangle$ in $\mathcal{A}$, we have $H(C) \bowtie H(D)$ or $H(C) \bowtie p$, respectively.

The arity k of the Hintikka trees is determined by the number of quantified concepts of the form $\exists r.C$ or $\forall r.C$ that appear in $\mathsf{sub}(\mathcal{O})$. Intuitively, each successor will act as the witness for one of these restrictions. We define $K := \{1, \ldots, k\}$ to be the index set of all successors. Since we need to know which successor in the tree is the witness of which restriction, we fix an arbitrary bijection

$$\varphi\colon \{D \mid D \in \mathsf{sub}(\mathcal{O}) \text{ is of the form } \exists r.C \text{ or } \forall r.C\} \to K.$$

For a given role name $r \in \mathsf{N_R}$, we denote by Φ_r the set of all indices $\varphi(D)$ where $D \in \mathsf{sub}(\mathcal{O})$ is a quantified concept of the form $\exists r.C$ or $\forall r.C$; i.e., all the indices that correspond to quantified concepts constructed using the role r.

The following condition ensures that the functions labeling the different successors of a node verify the semantics of the existential and value restrictions that appear in the ontology.

Definition 12 (Hintikka condition)**.** The tuple of $k+1$ Hintikka functions $(H_0, H_1, \ldots, H_k)$ satisfies the *Hintikka condition* if:

- for every existential restriction $\exists r.C \in \mathsf{sub}(\mathcal{O})$, the following hold:
  - $H_0(\exists r.C) = \min(H_i(\rho), H_i(C))$ for $i = \varphi(\exists r.C)$, and
  - $H_0(\exists r.C) \geq \min(H_j(\rho), H_j(C))$ for all $j \in \Phi_r$;
- for every value restriction $\forall r.C \in \mathsf{sub}(\mathcal{O})$, the following hold:
  - $H_0(\forall r.C) = H_i(\rho) \Rightarrow H_i(C)$ for $i = \varphi(\forall r.C)$, and
  - $H_0(\forall r.C) \leq H_j(\rho) \Rightarrow H_j(C)$ for all $j \in \Phi_r$.

We briefly explain the intuition behind this definition. For an existential restriction $\exists r.C$, the first condition makes sure that its interpretation is witnessed by the designated successor $\varphi(\exists r.C)$. The second condition additionally ensures that the degree of this existential restriction is indeed the supremum of the degrees of all r-successors. The conditions for value restrictions are dual, ensuring the appropriate witness,

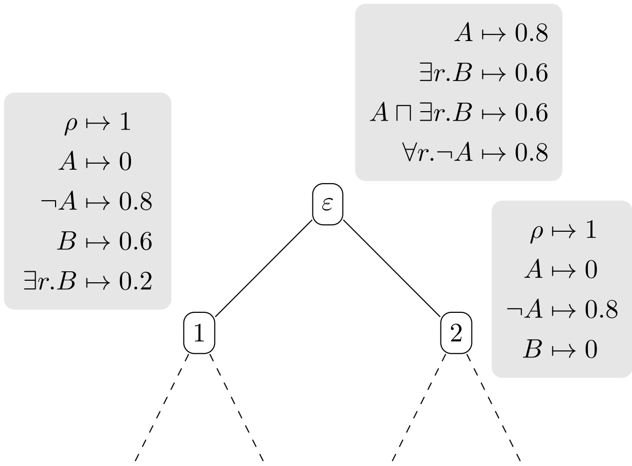

Figure 1: A Hintikka tree for the ontology from Example 13

and that this witness provides the infimum of the degrees among the r-successors. Notice that if $H_i(\rho) = 0$, then the inequalities in these conditions are trivially satisfied.

A *Hintikka tree* for the local ontology $\mathcal{O} = (\mathcal{A}, \mathcal{T})$ is an infinite k-ary tree $\mathbf{T}$ labeled with Hintikka functions for $\mathcal{O}$ that are compatible with $\mathcal{T}$ such that $\mathbf{T}(\varepsilon)$ is compatible with $\mathcal{A}$ and for every node $u \in K^*$, the tuple $(\mathbf{T}(u), \mathbf{T}(u1), \dots, \mathbf{T}(uk))$ satisfies the Hintikka condition. Compatibility ensures that all the terminological axioms are satisfied at any node of the Hintikka tree, while the Hintikka condition makes sure that the tree is in fact a (witnessed) model of the TBox.

Example 13. Consider the chain $L = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ and the ontology $\mathcal{O} = (\mathcal{A}, \mathcal{T})$, where

$$\mathcal{A} = \{\langle a\colon (A \sqcap \exists r.B) > 0.4\rangle\} \text{ and } \mathcal{T} = \{\langle A \sqsubseteq \forall r.\neg A\rangle, \langle B \sqsubseteq \exists r.B \geq 0.2\rangle\}.$$

The set $\mathsf{sub}(\mathcal{O})$ consists of the elements A, $\neg A$, B, $\exists r.B$, $A \sqcap \exists r.B$, and $\forall r.\neg A$. The arity of the Hintikka trees is 2 since the only quantified concepts are $\exists r.B$ and $\forall r.\neg A$. We fix the mapping φ by setting $\varphi(\exists r.B) := 1$ and $\varphi(\forall r.\neg A) := 2$.

Figure 1 depicts the initial part of a Hintikka tree $\mathbf{T}$ for $\mathcal{O}$, where the values that are not shown are implicitly 0. The Hintikka function $\mathbf{T}(\varepsilon)$ at the root must be compatible with $\mathcal{A}$, i.e. we must have $\mathbf{T}(\varepsilon)(A \sqcap \exists r.B) > 0.4$; here, this value was guessed to be 0.6. Definition 11 now forces us to assign to A and $\exists r.B$ two values whose minimum is 0.6, for example 0.8 and 0.6, respectively. The GCI $\langle A \sqsubseteq \forall r.\neg A\rangle$ requires the degree of $\mathbf{T}(\varepsilon)(\forall r.\neg A)$ to be ≥ 0.8. The other GCI does not constrain the initial Hintikka function any further.

The Hintikka function $\mathbf{T}(1)$ labeling the first successor must satisfy the first condition of Definition 12, which enforces that $\min(\mathbf{T}(1)(\rho), \mathbf{T}(1)(B))$ must be 0.6. If we guess $\mathbf{T}(1)(B) = 0.6$, then to achieve compatibility with $\mathcal{T}$, we need to set $\mathbf{T}(1)(\exists r.B)$ to at least 0.2. Finally, since $\mathbf{T}(\varepsilon)(\forall r.\neg A) = 0.8$, we must have $\mathbf{T}(1)(\neg A) = 0.8$ by the Hintikka condition, and thus $\mathbf{T}(1)(A) = 0$. The second successor can be constructed to witness the fact that $\mathbf{T}(\varepsilon)(\forall r.\neg A) = 0.8$ following similar arguments.

This example demonstrates that, using the notion of Hintikka trees, one can inductively build a model of the input ontology $\mathcal{O}$, if one exists. The proof of the following theorem uses arguments similar to those in [13]. The main difference is the presence of successors witnessing the value restrictions, and the condition at the root of the tree ensuring local consistency. The full proof follows the same steps as [16], where only the root condition needs to be extended to handle order assertions.

Lemma 14. *In L-$\mathfrak{N}\mathcal{AL}$, a local ontology $\mathcal{O} = (\mathcal{A}, \mathcal{T})$ is consistent iff there is a Hintikka tree for $\mathcal{O}$.*

By building looping automata whose runs correspond exactly to those Hintikka trees, we reduce consistency reasoning in L-$\mathfrak{N}\mathcal{AL}$ to the emptiness problem of such automata.

Definition 15 (Hintikka automaton)**.** Let $\mathcal{O} = (\mathcal{A}, \mathcal{T})$ be a local ontology. The *Hintikka automaton* for $\mathcal{O}$ is the LA $\mathbf{A}_\mathcal{O} = (Q, I, \Delta)$, where

- Q is the set of all Hintikka functions compatible with $\mathcal{T}$,
- I is the set of all Hintikka functions from Q that are compatible with the local ABox $\mathcal{A}$, and
- Δ is the set of all tuples from Q^{k+1} that satisfy the Hintikka condition.

The runs of $\mathbf{A}_\mathcal{O}$ are exactly those Hintikka trees $\mathbf{T}$ for $\mathcal{O}$ where $\mathbf{T}(\varepsilon)$ is compatible with all the assertions in $\mathcal{A}$. Thus, $\mathcal{O}$ is locally consistent iff $\mathbf{A}_\mathcal{O}$ is non-empty.

Recall that k is bounded by the number $|\mathsf{sub}(\mathcal{O})|$ of subconcepts appearing in $\mathcal{O}$, which is polynomial in the size of $\mathcal{O}$. Since there are at most $|L|^{|\mathsf{sub}(\mathcal{O})|+1}$ Hintikka functions, the size of Δ is bounded by $|L|^{(|\mathsf{sub}(\mathcal{O})|+1)^2}$, and hence the size of the automaton $\mathbf{A}_\mathcal{O}$ is exponential in the input. By Proposition 9, local consistency in L-$\mathfrak{NAL}$ can be decided in exponential time. Recall that deciding concept satisfiability w.r.t. general TBoxes is already ExpTime-hard for classical $\mathcal{ALC}$ [28], which is equivalent to $\mathfrak{NAL}$ with two-valued semantics. We can transform any classical $\mathcal{ALC}$ TBox $\mathcal{T}$ into a L-$\mathfrak{NAL}$ TBox $\mathcal{T}'$, for any finite set L, in which all GCIs are crisp; that is, of the form $\langle C \sqsubseteq D \geq 1\rangle$, such that the concept E is satisfiable w.r.t. $\mathcal{T}$ iff the ontoloy $(\{\langle a : E \geq 1\rangle\}, \mathcal{T}')$ is consistent. We thus obtain a tight complexity bound for local consistency, as formalized in the following theorem.

Theorem 16. *Deciding local consistency in L-$\mathfrak{NAL}$ w.r.t. general TBoxes is* ExpTime*-complete.*

In the next section we show that the same complexity bounds hold if we allow the whole interval $[0, 1]$ in the semantics. However, we need to generalize the notion of Hintikka trees in such a way that they represent classes of models, rather than just single models of the ontology.

4.2. Local Consistency in G-$\mathfrak{NAL}$

It is a simple observation that any set of truth values that contains 0 and 1 is closed w.r.t. the Gödel connectives. Owing to this observation, it is common to restrict reasoning in fuzzy DLs with Gödel semantics to the finitely many truth values occurring in the ontology [29, 30]. This restriction is also sometimes justified by the “limited precision of computers” [31].

However, the restriction to finitely many truth degrees affects the expressivity of the language, even for the very simple description logic G-$\mathcal{AL}$, the sublogic of G-$\mathfrak{NAL}$ which allows $\exists$, $\forall$, $\sqcap$, and $\top$, but no negations $\neg$. It is possible to show an even stronger result: reasoning in this logic cannot, without loss of generality, be restricted to *finitely-valued* models, i.e., models that only use values from an arbitrary finite subset of $[0, 1]$. Notice that for the related *Zadeh semantics*, which differ from the Gödel semantics only in the operators used to interpret implication and negation, reasoning can be restricted to finitely-valued models without loss of generality [4, 21].

Example 17. Let $\mathcal{T}_0$ be the G-$\mathcal{AL}$ TBox

$$\mathcal{T}_0 = \{\langle \forall r.A \sqsubseteq A\rangle,\ \langle \exists r.\top \sqsubseteq A\rangle\}.$$

We show that $\top$ is *not* 1-subsumed by A w.r.t. the ontology $\mathcal{O} = (\emptyset, \mathcal{T}_0)$, but every finitely-valued model of this ontology also satisfies $\langle \top \sqsubseteq A\rangle$. For the former, we construct a model $\mathcal{I}_0$ of $\mathcal{T}_0$ as follows (see Figure 2). Let $\Delta^{\mathcal{I}_0}$ be the set of all positive natural numbers. We define $A^{\mathcal{I}_0}(n) := r^{\mathcal{I}_0}(n, n+1) := \frac{1}{n+1}$ for all $n \in \mathbb{N}$ and $r^{\mathcal{I}_0}(n, m) := 0$ if $m \neq n + 1$. It is straightforward to check that this is a witnessed model of $\mathcal{T}_0$ that violates $\langle \top \sqsubseteq A\rangle$. Thus, $\top$ is not 1-subsumed by A w.r.t. $\mathcal{O}$. In fact, the best subsumption degree of $\top$ and A w.r.t. $\mathcal{O}$ is 0.

Assume now that there is a witnessed model $\mathcal{I}$ of $\mathcal{T}_0$ using only finitely many truth values that violates $\langle \top \sqsubseteq A\rangle$. Since $\mathcal{I}$ uses only finitely many truth values, there exists an element $y \in \Delta^\mathcal{I}$ for which $A^\mathcal{I}(y)$ is minimal, that is, $A^\mathcal{I}(y) \leq A^\mathcal{I}(x)$ holds for all $x \in \Delta^\mathcal{I}$. Furthermore, since $\mathcal{I}$ violates $\langle \top \sqsubseteq A\rangle$ there must be some $x_0 \in \Delta^\mathcal{I}$ satisfying $A^\mathcal{I}(x_0) < 1$. In particular, this yields $A^\mathcal{I}(y) < 1$.

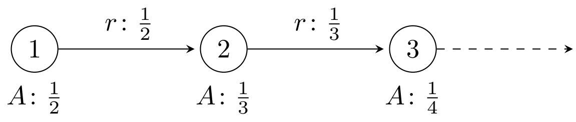

Figure 2: The model $\mathcal{I}_0$ from Example 17

As $\mathcal{I}$ is witnessed, there must exist a $z \in \Delta^{\mathcal{I}}$ such that $(\forall r.A)^{\mathcal{I}}(y) = r^{\mathcal{I}}(y,z) \Rightarrow A^{\mathcal{I}}(z)$. The first axiom of $\mathcal{T}_0$ entails $r^{\mathcal{I}}(y,z) \Rightarrow A^{\mathcal{I}}(z) \leq A^{\mathcal{I}}(y) < 1$, and in particular

$$r^{\mathcal{I}}(y,z) > A^{\mathcal{I}}(z). \tag{1}$$

The second axiom from $\mathcal{T}_0$ yields

$$r^{\mathcal{I}}(y,z) = \min(r^{\mathcal{I}}(y,z), 1) \leq (\exists r.\top)^{\mathcal{I}}(y) \leq A^{\mathcal{I}}(y). \tag{2}$$

From (1) and (2) we obtain $A^{\mathcal{I}}(y) > A^{\mathcal{I}}(z)$, contradicting the minimality of $A^{\mathcal{I}}(y)$. We have thus shown that a witnessed model of $\mathcal{T}_0$ with only finitely many truth values cannot violate $\langle \top \sqsubseteq A \rangle$. In particular, $\mathcal{T}_0$ entails $\langle \top \sqsubseteq A \rangle$ when reasoning is restricted to a finite set of truth degrees.

Recall that a (fuzzy) DL has the *finite model property* if every consistent ontology has a model with finite domain. A simple consequence of the last example is that G-$\mathcal{AL}$ does not have the finite model property. Indeed, $\mathcal{I}_0$ is a model of the ontology $(\{\langle a{:}A = 0.5 \rangle\}, \mathcal{T}_0)$ if we interpret the individual name a as $a^{\mathcal{I}_0} := 1$. This shows that this ontology is consistent. However, any finite model $\mathcal{I}$ of $\mathcal{T}_0$ uses only finitely many truth degrees. As shown in the example, such an interpretation must satisfy $A^{\mathcal{I}}(x) = 1$ for all $x \in \Delta^{\mathcal{I}}$, and hence violate the assertion $\langle a{:}A = 0.5 \rangle$. We thus obtain the following result.

Theorem 18. *G-$\mathcal{AL}$ does not have the finite model property nor the finitely-valued model property.*

The lack of the finitely-valued model property implies that some of the standard techniques used for reasoning in fuzzy DLs cannot be directly applied to any logic that contains G-$\mathcal{AL}$. For example, termination of the tableaux-based approach [32, 33] relies on the existence of finitely many *types* that can describe domain elements by specifying the membership degrees for all relevant concepts, while any sound and complete reduction to crisp reasoning [4, 29] implies the finitely-valued model property. We also cannot directly use the construction from Section 4.1, as it would result in an automaton with infinitely many states.

Since all known undecidability proofs for fuzzy DLs [7–10, 34] are based on the fact that one can enforce models to have infinitely many values, one could thus be inclined to believe that consistency in G-$\mathfrak{N}\mathcal{AL}$ is also undecidable. We will show that this is not the case, and in fact consistency, subsumption, and satisfiability are ExpTime-complete. For now we focus on local consistency.

Our approach for handling infinitely many truth degrees is based on the observation that the axioms and the semantics of the constructors only introduce restrictions on the *order* of the values that models can assign to concepts, not on the values themselves. For example, an interpretation $\mathcal{I}$ satisfies an assertion $\langle a{:}(A \sqcap B) \geq p \rangle$ iff $A^{\mathcal{I}}(a^{\mathcal{I}}) \geq p$ and $B^{\mathcal{I}}(a^{\mathcal{I}}) \geq p$. Thus, rather than building a model directly, we first create an abstract representation of this model by encoding, for each domain element, only the order between concepts. A similar approach has been previously used for fuzzy extensions of other logics based on Gödel semantics [35, 36]. Here we show how the abstraction to ordering affects the construction of Hintikka trees and Hintikka automata recognizing them.

Example 19. Consider again the TBox $\mathcal{T}_0$ from Example 17. When trying to construct a model violating $\langle \top \sqsubseteq A \rangle$, we start with a domain element satisfying the restriction that the value of A is strictly smaller than 1 (see Figure 3). The second axiom implies that the degree of any outgoing r-connection is bounded by the value of A. Moreover, the first axiom states that the witness of $\forall r.A$ must satisfy A to a degree strictly smaller than the value of the r-connection, and thus strictly smaller than the original value of A.

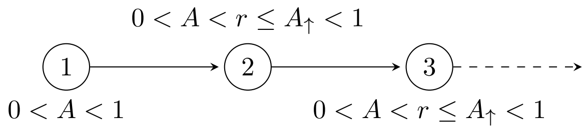

Figure 3: An abstract description of $\mathcal{I}_1$ from Example 17

This yields an abstract description of two domain elements in terms of order relations between values of concepts at the current node and the parent node (denoted by a subscript $\uparrow$). Applying the same argument to the new element yields another element with the same restrictions. In order to construct a model, it is easy to see that the value of A at all considered elements has to be strictly greater than 0—once the value of A is 0, there can be no successors with smaller values for A. Note that it suffices to consider order relations between concepts of neighboring elements, which are directly connected by some role to a degree greater than 0. In other words, the construction of this abstract model can be done locally.

As in this example, we use the subscript $\uparrow$ to refer to values of the parent node in the tree-like model that we will construct. As in Section 4.1, we use ρ to represent the degree of the role connection from the parent node.

Definition 20 (order structure $\mathcal{U}$)**.** We define the set $\mathsf{sub}_\uparrow(\mathcal{O}) := \{C_\uparrow \mid C \in \mathsf{sub}(\mathcal{O})\}$ and the set $\mathcal{U} := \mathcal{V}_\mathcal{O} \cup \mathsf{sub}(\mathcal{O}) \cup \mathsf{sub}_\uparrow(\mathcal{O}) \cup \{\rho\}$. Let order$(\mathcal{U})$ denote the set of all total preorders $\lesssim_*$ over $\mathcal{U}$ that have 0 and 1 as least and greatest element, respectively, and preserve the order of real numbers on $\mathcal{V}_\mathcal{O} = \mathcal{U} \cap [0,1]$. For any $\lesssim_* \in$ order$(\mathcal{U})$ and $\bowtie \in \{=, \geq, >, \leq, <\}$, we denote by $\bowtie_*$ the corresponding relation induced by the total preorder $\lesssim_*$, i.e. $\equiv_*$, $\gtrsim_*$, $>_*$, $\lesssim_*$, or $<_*$, which are defined as usual.

Given $\lesssim_* \in$ order$(\mathcal{U})$, the following functions on $\mathcal{U}$ that mimic the operators of Gödel fuzzy logic over $[0,1]$ are well-defined since $\lesssim_*$ is total:

$$\min\nolimits_*(x,y) := \begin{cases} x & \text{if } x \lesssim_* y \\ y & \text{otherwise,} \end{cases}$$

$$\operatorname{res}_*(x,y) := \begin{cases} 1 & \text{if } x \lesssim_* y \\ y & \text{otherwise.} \end{cases}$$

It is easy to see that the operators $\min_*$ and res_* agree with min and $\Rightarrow$, respectively, on the set $\mathcal{V}_\mathcal{O}$. We use the subscript $*$ to distinguish different total preorders from order$(\mathcal{U})$.

For convenience, we extend the notation of $\mathsf{sub}_\uparrow(\mathcal{O})$ to the elements of $\mathcal{V}_\mathcal{O}$ by setting $q_\uparrow := q$ for all $q \in \mathcal{V}_\mathcal{O}$. Using the total preorders from order$(\mathcal{U})$, we can now describe the relationships between all the subconcepts from $\mathcal{O}$ and the truth degrees from $\mathcal{V}_\mathcal{O}$ at given domain elements. One can think of such a preorder as the *type* of a domain element, from which a tree-shaped interpretation can be built.

As for the finitely-valued case, let k be the number of quantified concepts in $\mathsf{sub}(\mathcal{O})$ and φ an arbitrary but fixed bijection between the set of all quantified concepts in $\mathsf{sub}(\mathcal{O})$ and $K := \{1, \ldots, k\}$. This bijection specifies which quantified concept is witnessed by which successor in the tree. The following definition lifts the notion of Hintikka functions to the new setting (cf. Definition 11).

Definition 21 (Hintikka ordering)**.** A *Hintikka ordering* is a total preorder $\lesssim_H \in$ order$(\mathcal{U})$ that satisfies the following conditions for every $C \in \mathsf{sub}(\mathcal{O})$:

- $C = \top$ implies $C \equiv_H 1$,
- if $C = D_1 \sqcap D_2$, then $C \equiv_H \min_H(D_1, D_2)$, and
- if $C = \neg D$, then $C \equiv_H \operatorname{res}_H(D, 0)$.

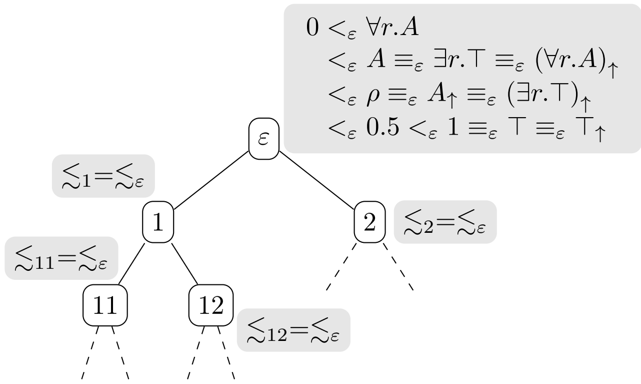

Figure 4: A Hintikka tree for Example 17

This preorder is *compatible* with the TBox $\mathcal{T}$ if for every GCI $\langle C \sqsubseteq D \geq p\rangle \in \mathcal{T}$ we have $\mathrm{res}_H(C, D) \gtrsim_H p$. It is *compatible* with $\mathcal{A}$ if for every order assertion $\langle a : C \bowtie p\rangle$ or $\langle a : C \bowtie a : D\rangle$ in $\mathcal{A}$, we have $C \bowtie_H p$ or $C \bowtie_H D$, respectively.

The conditions imposed on Hintikka orderings ensure that they preserve the semantics of all the *propositional* constructors. For every *quantified* concept, we still need to ensure the existence of a witness. This is achieved through φ and the following adapted Hintikka condition (cf. Definition 12).

Definition 22 (Ordered Hintikka condition)**.** The *(ordered) Hintikka condition* consists of the following requirements for a $(k{+}1)$-tuple $(\lesssim_0, \lesssim_1, \dots, \lesssim_k)$ of Hintikka orderings:

- for every $1 \leq i \leq k$ and all $\alpha, \beta \in \mathcal{V}_\mathcal{O} \cup \mathsf{sub}(\mathcal{O})$, we have $\alpha \lesssim_0 \beta$ iff $\alpha_\uparrow \lesssim_i \beta_\uparrow$;
- for every $\exists r.C \in \mathsf{sub}(\mathcal{O})$, we have
  - $(\exists r.C)_\uparrow \equiv_i \min_i(\rho, C)$ for $i = \varphi(\exists r.C)$, and
  - $(\exists r.C)_\uparrow \gtrsim_i \min_i(\rho, C)$ for all $i \in \Phi_r$; and
- for every $\forall r.C \in \mathsf{sub}(\mathcal{O})$, we have
  - $(\forall r.C)_\uparrow \equiv_i \mathrm{res}_i(\rho, C)$ for $i = \varphi(\forall r.C)$, and
  - $(\forall r.C)_\uparrow \lesssim_i \mathrm{res}_i(\rho, c)$ for all $i \in \Phi_r$.

An *(ordered) Hintikka tree* for $\mathcal{O}$ is an infinite k-ary tree, where every node u is associated with a Hintikka ordering $\lesssim_u$ compatible with $\mathcal{T}$, such that:

- every tuple $(\lesssim_u, \lesssim_{u1}, \dots, \lesssim_{uk})$ satisfies the Hintikka condition, and
- $\lesssim_\varepsilon$ is compatible with $\mathcal{A}$.

For instance, Figure 4 depicts a Hintikka tree for the TBox $\mathcal{T}_0$ from Example 17 and the local ABox $\mathcal{A} = \{\langle a : A < 1\rangle\}$. Notice that every node is labeled with the same preorder and the tree is invariant w.r.t. the choice of φ. As in the finitely-valued case, the existence of an (ordered) Hintikka tree for a local ontology $\mathcal{O}$ implies the consistency of $\mathcal{O}$. Conversely, every model can be transformed into a Hintikka tree. The idea is to *unravel* the model into an infinite tree by creating copies of domain elements to remove any cycles, and then abstract from the specific values by just considering the ordering between the elements of $\mathcal{U}$. Overall, we obtain the following characterization in analogy to that of Lemma 14.

Lemma 23. *In* G-$\mathfrak{N}\mathcal{AL}$*, a local ontology* $\mathcal{O}$ *is consistent iff there is an ordered Hintikka tree for* $\mathcal{O}$*.*

We can now proceed to define a new kind of Hintikka automaton whose runs are exactly the ordered Hintikka trees for $\mathcal{O}$ (cf. Definition 15).

Definition 24. The *(ordered) Hintikka automaton* for a local ontology $\mathcal{O}$ is the LA $\mathbf{A}_\mathcal{O} := (Q, I, \Delta)$, where

- Q is the set of all Hintikka orderings compatible with $\mathcal{T}$,
- I is the set of all Hintikka orderings from Q that are compatible with $\mathcal{A}$, and
- Δ contains all tuples from Q^{k+1} that satisfy the Hintikka condition.

Observe that the number of Hintikka orderings for $\mathcal{O}$ is bounded by $2^{|\mathcal{U}|^2}$ and the cardinality of the set $\mathcal{U} = \mathcal{V}_\mathcal{O} \cup \mathsf{sub}(\mathcal{O}) \cup \mathsf{sub}_\uparrow(\mathcal{O}) \cup \{\rho\}$ is linear in the size of $\mathcal{O}$. Likewise, the arity k of the automaton is bounded by $|\mathsf{sub}(\mathcal{O})|$, which is linear in the size of $\mathcal{O}$. Thus, the size of the Hintikka automaton $\mathbf{A}_\mathcal{O}$ is exponential in the size of $\mathcal{O}$. By Proposition 9, we again obtain an ExpTime-decision procedure for consistency of ontologies with local ordered ABoxes in G-$\mathfrak{N}\mathcal{AL}$. Reasoning in G-$\mathfrak{N}\mathcal{AL}$ with only assertions of the form $\langle \alpha \geq p \rangle$ is equivalent to reasoning in classical $\mathcal{ALC}$ [10], which is already ExpTime-hard w.r.t. general TBoxes [28]. Hence, our complexity bounds are tight.

Theorem 25. *Deciding local consistency in* G-$\mathfrak{N}\mathcal{AL}$ *w.r.t. general TBoxes is* ExpTime*-complete.*

In the following section, we remove the restriction to local ordered ABoxes and show that consistency, as well as the other reasoning problems considered in this paper, remains ExpTime-complete.

4.3. Consistency and Other Reasoning Problems

We have so far shown how to decide consistency of an ontology under the assumption that the ABox is local; i.e., it refers to the knowledge of one individual name only. In general, ontologies contain concept assertions about several different individuals, and these individuals are also related via role assertions. To handle this general case, we adapt a technique from classical DLs known as *pre-completion* [37].

Recall that every consistent local ontology has a tree-shaped model, with the root node interpreting the (unique) individual name appearing in the ABox. Generalizing this idea, it is easy to see that every consistent ontology has a *forest-shaped* model; that is, a model composed of a finite set of trees, with their roots interpreting the individuals of the ABox, whose roots can be arbitrarily interconnected through roles. The idea of pre-completion is to extend the input ABox to a full specification of the degrees to which each individual belongs to each of the relevant concepts of the ontology. Clearly, while this can be done explicitly for the finitely-valued case, in G-$\mathfrak{N}\mathcal{AL}$ it is necessary to abstract from the specific membership degrees and specify only a preorder among them, as done for local consistency. This is achieved using order assertions. In the following, we provide a general definition of pre-completion that can be used to reduce consistency to local consistency in both, the finite and the infinitely-valued cases. For this definition, we require the notion of a coherent ordered ABox. An ordered ABox $\mathcal{A}$ is *coherent* if it is possible to map every crisp assertion appearing in $\mathcal{A}$ to an element in $\mathfrak{C}$, such that all the order assertions in $\mathcal{A}$ are satisfied.

Definition 26 (pre-completion)**.** Let $\mathcal{O} = (\mathcal{A}, \mathcal{T})$ be an ontology, and let $\mathsf{Ind}(\mathcal{A})$ denote the set of all individual names that appear in the ABox $\mathcal{A}$, $\mathsf{Role}(\mathcal{O})$ the set of all role names appearing in $\mathcal{O}$, and $\mathsf{As}(\mathcal{O}) := \{a\!:\!C, (a,b)\!:\!r \mid a, b \in \mathsf{Ind}(\mathcal{A}), C \in \mathsf{sub}(\mathcal{O}), r \in \mathsf{Role}(\mathcal{O})\}$ the set of all crisp assertions that may be relevant for $\mathcal{O}$. A *pre-completion* of $\mathcal{A}$ w.r.t. $\mathcal{T}$ is a coherent ordered ABox $\mathcal{B}$ such that:

1. $\mathcal{A} \subseteq \mathcal{B}$;
2. for every $\alpha \in \mathsf{As}(\mathcal{O})$ and $\beta \in \mathsf{As}(\mathcal{O}) \cup \mathcal{V}_\mathcal{O}$ we have either $\langle \alpha < \beta \rangle$, $\langle \alpha = \beta \rangle$, or $\langle \alpha > \beta \rangle$ in $\mathcal{B}$;
3. for every $a \in \mathsf{Ind}(\mathcal{A})$ and all $C \in \mathsf{sub}(\mathcal{O})$:
   - if $C = \top$, then $\langle a\!:\!C = 1 \rangle \in \mathcal{B}$,
   - if $C = D_1 \sqcap D_2$, then if $\langle a\!:\!D_1 < a\!:\!D_2 \rangle \in \mathcal{B}$, then $\langle a\!:\!C = a\!:\!D_1 \rangle \in \mathcal{B}$; and $\langle a\!:\!C = a\!:\!D_2 \rangle \in \mathcal{B}$ otherwise,
   - if $C = \neg D$, then if $\langle a\!:\!D > 0 \rangle \in \mathcal{B}$, then $\langle a\!:\!C = 0 \rangle \in \mathcal{B}$; otherwise, $\langle a\!:\!C = 1 \rangle \in \mathcal{B}$;

4. for every $\exists r.C \in \mathsf{sub}(\mathcal{O})$ and $a, b \in \mathsf{Ind}(\mathcal{A})$, $\{\langle a{:}\exists r.C \geq (a,b){:}r\rangle, \langle a{:}\exists r.C \geq b{:}C\rangle\} \cap \mathcal{B} \neq \emptyset$;

5. for every $\forall r.C \in \mathsf{sub}(\mathcal{O})$ and $a, b \in \mathsf{Ind}(\mathcal{A})$, $\{\langle a{:}\forall r.C \leq b{:}C\rangle, \langle (a,b){:}r \leq b{:}C\rangle\} \cap \mathcal{B} \neq \emptyset$; and

6. for all $a \in \mathsf{Ind}(\mathcal{A})$ and every GCI $\langle C \sqsubseteq D \geq p\rangle \in \mathcal{T}$, $\{\langle a{:}C \leq a{:}D\rangle, \langle a{:}D \geq p\rangle\} \cap \mathcal{B} \neq \emptyset$.

This definition generalizes the local conditions imposed by Hintikka functions and Hintikka orderings to handle several individuals simultaneously. Notice that these conditions do not guarantee the existence of witnesses for the quantified concepts, but only that they are not violated by the role assertions. From such a pre-completion we can generate a set of local ontologies whose consistency tests will handle the witnessing of quantified concepts.

Notice that the pre-completion $\mathcal{B}$ has polynomially many order assertions. It can be seen as a system of linear inequalities between the "variables" in $\mathsf{As}(\mathcal{O})$. In the case of the Gödel t-norm, these inequalities refer to the whole interval $[0, 1]$, and hence their consistency is well-known to be decidable in polynomial time [38, 39]. For finite chains, it requires to solve an *integer* linear program, which can be done in non-deterministic polynomial time [40]. Given a pre-completion $\mathcal{B}$ and $a \in \mathsf{Ind}(\mathcal{A})$, we define the local ordered ABox $\mathcal{B}_a$ as the restriction of $\mathcal{B}$ to only order restrictions on concept assertions over a; that is,

$$\mathcal{B}_a := \{\langle a{:}C \bowtie p\rangle \in \mathcal{B}\} \cup \{\langle a{:}C \bowtie a{:}D\rangle \in \mathcal{B}\}.$$

Lemma 27. *An ontology $\mathcal{O} = (\mathcal{A}, \mathcal{T})$ is consistent iff there is a pre-completion $\mathcal{B}$ of $\mathcal{A}$ w.r.t. $\mathcal{T}$ such that, for every $a \in \mathsf{Ind}(\mathcal{A})$, the local ontology $\mathcal{O}_a := (\mathcal{B}_a, \mathcal{T})$ is consistent.*

Notice that every pre-completion must be a set of order assertions between elements of $\mathsf{As}(\mathcal{O}) \cup \mathcal{V}_{\mathcal{O}}$. There are exponentially many such sets, each of which is of size polynomial in the size of $\mathcal{O}$. To decide consistency of $\mathcal{O}$, we can enumerate all candidate ordered ABoxes $\mathcal{B}$ and check for each of them that (i) it satisfies all the conditions from Definition 26 (in non-deterministic polynomial time, and hence also in deterministic exponential time), and (ii) each of the local ontologies $\mathcal{O}_a$ is consistent (in exponential time; see Theorems 16 and 25). Overall, we obtain the following complexity result.

Theorem 28. *Deciding consistency in $\mathfrak{C}$-$\mathfrak{N}\mathcal{AL}$ w.r.t. general TBoxes is* ExpTime*-complete.*

We now turn our attention to the remaining reasoning problems introduced in Definition 3; namely, deciding satisfiability and subsumption between concepts, as well as computing the best degrees to which these entailments hold. Recall that, as described in Section 2, for these reasoning problems we can assume w.l.o.g. that the ABox is empty.

Let $\mathcal{O} = (\emptyset, \mathcal{T})$ be an ontology. It is easy to see that p-subsumption and p-satisfiability w.r.t. $\mathcal{O}$ can be reduced in polynomial time to consistency w.r.t. local ordered ABoxes. More precisely, for any two concepts C, D and $p \in [0, 1]$,

- C is p-satisfiable w.r.t. $\mathcal{O}$ iff $(\{\langle a{:}C \geq p\rangle\}, \mathcal{T})$ is consistent, and
- C is p-subsumed by D w.r.t. $\mathcal{O}$ iff $(\{\langle a{:}D < p\rangle, \langle a{:}C > a{:}D\rangle\}, \mathcal{T})$ is inconsistent,

where a is an arbitrary individual name. We thus obtain the following result from Theorem 25.

Theorem 29. *The problem of deciding satisfiability or subsumption in $\mathfrak{C}$-$\mathfrak{N}\mathcal{AL}$ w.r.t. general TBoxes is* ExpTime*-complete.*

Consider now the problems of computing the *best* satisfiability and subsumption degrees. In the case that $\mathfrak{C}$ is a finite set $L \subseteq [0, 1]$, then these best entailment degrees can be found by verifying, for every $p \in L$, whether p-satisfiability or p-subsumption holds, respectively. Overall, this requires $|L|$ entailment tests, each of which can be solved in exponential time. Thus, the best satisfiability and subsumption degrees w.r.t. an L-$\mathfrak{N}\mathcal{AL}$ ontology can be computed in exponential time.

Obviously, the computation procedure sketched above cannot be performed in G-$\mathfrak{N}\mathcal{AL}$, as it would require uncountably many calls to the decision procedure. To obtain a terminating procedure, we exploit once again the idea of using preorders from order($\mathcal{U}$) and Hintikka trees to show that the local consistency checks required for deciding p-satisfiability and p-subsumption only depend on the position of p relative to the values occurring in $\mathcal{T}$, but not on the precise value of p.

Lemma 30. *Let p, p' be elements of the open interval (p_i, p_{i+1}) for two adjacent values $p_i, p_{i+1} \in \mathcal{V}_\mathcal{O}$, and C, D be concepts. Then $(\{\langle a{:}C \geq p\rangle\}, \mathcal{T})$ is consistent iff $(\{\langle a{:}C \geq p'\rangle\}, \mathcal{T})$ is consistent. Likewise, $(\{\langle a{:}D < p\rangle, \langle a{:}C > a{:}D\rangle\}, \mathcal{T})$ is consistent iff $(\{\langle a{:}D < p'\rangle, \langle a{:}C > a{:}D\rangle\}, \mathcal{T})$ is consistent.*

This lemma states that subsumption between C and D or satisfiability of C either holds for all values in an interval (p_i, p_{i+1}), or for none of them. Thus, there is no need to test them all for subsumption; choosing a representative of each interval, along with the elements of $\mathcal{V}_\mathcal{O}$, suffices. This yields the following result.

Corollary 31. *For any two concepts C and D, the best subsumption degree of C and D w.r.t. $\mathcal{O}$ and the best satisfiability degree of C w.r.t. $\mathcal{O}$ are always in $\mathcal{V}_\mathcal{O}$.*

Since the best subsumption degree p of C and D is always a subsumption degree, i.e., C is p-subsumed by D, it suffices to check subsumption w.r.t. the values from $\mathcal{V}_\mathcal{O}$ in order to determine the best subsumption degree. Thus, we only have to execute linearly many (in-)consistency checks to compute the best subsumption degree.

However, it is possible that C is p-satisfiable for every $p \in (p_i, p_{i+1})$, but not p_{i+1}-satisfiable. Therefore, we check satisfiability for all values $\frac{p_i + p_{i+1}}{2}$. The best satisfiability degree is then the largest p_{i+1} for which this check succeeds (or 0 if it never succeeds). Again, this means that we have to execute linearly many consistency checks to compute the best satisfiability degree. By combining these reductions with Theorems 16 and 25, we obtain the following results.

Corollary 32. *In $\mathfrak{C}$-$\mathfrak{N}\mathcal{AL}$, best subsumption and satisfiability degrees w.r.t. general TBoxes can be computed in exponential time.*

This finishes our study of reasoning problems for $\mathfrak{C}$-$\mathfrak{N}\mathcal{AL}$. In particular, we have shown that also subsumption in $\mathfrak{C}$-$\mathcal{FL}_0$ w.r.t. general TBoxes is ExpTime-complete since it is ExpTime-hard already for classical $\mathcal{FL}_0$ [41]. In the next section, we focus on the restricted case of subsumption in $\mathfrak{C}$-$\mathcal{FL}_0$ w.r.t. cyclic TBoxes under greatest fixed-point semantics, which is PSpace-complete for classical $\mathcal{FL}_0$ [11].

5. GFP-Subsumption w.r.t. Cyclic $\mathfrak{C}$-$\mathcal{FL}_0$ TBoxes

Our goal is to describe the restrictions imposed by a cyclic TBox $\mathcal{T}$ in normal form using automata. To simplify the construction, it is helpful to consider a generalization of WA where the transitions are associated with a weight from $\mathfrak{C}$.

Definition 33 (WWA)**.** A *weighted automaton with word transitions (WWA)* is a tuple of the form $\mathbf{A} = (\Sigma, Q, q_0, \mathsf{wt}, q_f)$, where

- Σ is a finite alphabet of *input symbols*,
- Q is a finite set of *states*,
- $q_0 \in Q$ is the *initial state*,
- $\mathsf{wt}\colon Q \times \Sigma^* \times Q \to \mathfrak{C}$ is the *transition weight function* with the property that its *support*

$$\mathsf{supp}(\mathsf{wt}) := \{(q, w, q') \in Q \times \Sigma^* \times Q \mid \mathsf{wt}(q, w, q') > 0\}$$

  is finite, and
- $q_f \in Q$ is the *final state*.

The notions of finite paths and labels are defined as for WA, and the set of all finite paths with label w in $\mathbf{A}$ is denoted by $\mathsf{paths}(\mathbf{A}, w)$. The *weight* of a finite path $\pi = q_0 w_1 q_1 w_2 \ldots w_n q_n$ in $\mathbf{A}$ is computed as $\mathsf{wt}(\pi) := \min_{i=1}^n \mathsf{wt}(q_{i-1}, w_i, q_i)$. The *behavior* $\|\mathbf{A}\|\colon \Sigma^* \to \mathfrak{C}$ of $\mathbf{A}$ is defined for all $w \in \Sigma^*$ by

$$\|\mathbf{A}\|(w) := \sup_{\pi \in \mathsf{paths}(\mathbf{A}, w)} \mathsf{wt}(\pi).$$

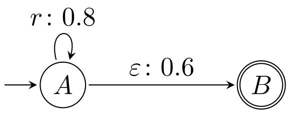

Figure 5: A weighted automaton constructed from $\mathcal{T}$ in Example 36

Intuitively, WWA generalize WA by adding a weight to each transition that the automaton needs to make during its execution. The total weight of a path is the minimum of the weights of the transitions it uses. This can be seen as a conjunction, since all these transitions need to be traversed in order to obtain the path. In WA, a word w is accepted if there is at least one accepting path for w; this is generalized in WWA to computing the maximum weight over all the possible paths for w. We can see the behavior of $\mathbf{A}$ as a fuzzy language on the alphabet Σ, where every word is given a membership degree from $\mathfrak{C}$.

Given a $\mathfrak{C}$-$\mathcal{FL}_0$ TBox $\mathcal{T}$, let $\mathsf{N}_\mathsf{C}^\mathcal{T}$ be the set of all concept names appearing in $\mathcal{T}$; that is, $\mathsf{N}_\mathsf{C}^\mathcal{T} := \mathsf{N}_\mathsf{P}^\mathcal{T} \cup \mathsf{N}_\mathsf{D}^\mathcal{T}$. We define a family of WWA, one for each pair of concept names in $\mathsf{N}_\mathsf{C}^\mathcal{T}$, as follows.

Definition 34 (automata $\mathbf{A}_{A,B}^\mathcal{T}$)**.** For concept names $A, B \in \mathsf{N}_\mathsf{C}^\mathcal{T}$, the WWA $\mathbf{A}_{A,B}^\mathcal{T} = (\mathsf{N}_\mathsf{R}, \mathsf{N}_\mathsf{C}^\mathcal{T}, A, \mathsf{wt}_\mathcal{T}, B)$ is defined by the transition weight function

$$\mathsf{wt}_\mathcal{T}(A', w, B') := \begin{cases} p & \text{if } \langle A' \sqsubseteq \forall w.B' \geq p \rangle \in \mathcal{T}, \\ 0 & \text{otherwise.} \end{cases}$$

Notice that for a given TBox $\mathcal{T}$ and concept names $A, A', B, B' \in \mathsf{N}_\mathsf{C}^\mathcal{T}$, the automata $\mathbf{A}_{A,B}^\mathcal{T}$ and $\mathbf{A}_{A',B'}^\mathcal{T}$ differ only in their initial and final states; the sets of states and transition weight function are always the same. Since $\mathcal{T}$ is in normal form, for any two concept names $A', B' \in \mathsf{N}_\mathsf{C}^\mathcal{T}$ and $w \in \mathsf{N}_\mathsf{R}^*$, there is at most one axiom $\langle A' \sqsubseteq \forall w.B' \geq p \rangle$ in $\mathcal{T}$, and hence the transition weight function is well-defined. This function has finite support since $\mathcal{T}$ is finite.

Using these automata, we can characterize the class of gfp-models of the TBox $\mathcal{T}$, as stated by the following lemmata.

Lemma 35. *For every gfp-model $\mathcal{I} = (\Delta, \cdot^\mathcal{I})$ of $\mathcal{T}$, $x \in \Delta$, and $A \in \mathsf{N}_\mathsf{C}^\mathcal{T}$,*

$$A^\mathcal{I}(x) = \inf_{B \in \mathsf{N}_\mathsf{P}^\mathcal{T}} \inf_{w \in \mathsf{N}_\mathsf{R}^*} \big(\|\mathbf{A}_{A,B}^\mathcal{T}\|(w) \Rightarrow (\forall w.B)^\mathcal{I}(x) \big).$$

This equation expresses the interpretation of a concept name A in terms of the behavior of the automata $\mathbf{A}_{A,B}^\mathcal{T}$ and the interpretations of value restrictions over the primitive concept names B, which only depend on the primitive interpretation $\mathcal{J}$ underlying $\mathcal{I}$. For primitive concept names A, the right-hand side is always equal to

$$\big(\|\mathbf{A}_{A,A}^\mathcal{T}\|(\varepsilon) \Rightarrow (\forall \varepsilon.A)^\mathcal{I}(x) \big) = \big(1 \Rightarrow A^\mathcal{J}(x) \big) = A^\mathcal{J}(x),$$

as expected. For defined concept names A, the intuition is that every finite path π in the automata $\mathbf{A}_{A,*}^\mathcal{T}$ corresponds to a conjunct of the form $\forall w.B$ in the (expanded) definition of A w.r.t. $\mathcal{T}$, where w is the label of π and $B \in \mathsf{N}_\mathsf{P}^\mathcal{T}$ is the final state of π. According to Definition 34, $\mathsf{wt}(\pi)$ then describes the degree to which $\forall w.B$ contributes to the interpretation of A. Note that, due to cycles in $\mathcal{T}$, in general there may be infinitely many concepts of the form $\forall w.B$ with $w \in \Sigma^*$, $B \in \mathsf{N}_\mathsf{P}^\mathcal{T}$ in the expanded interpretation of A.

Example 36. Consider the cyclic TBox in normal form

$$\mathcal{T} := \{ \langle A \sqsubseteq \forall \varepsilon.B \geq 0.6 \rangle, \langle A \sqsubseteq \forall r.A \geq 0.8 \rangle \}.$$

Figure 5 depicts the resulting WWA $\mathbf{A}_{A,B}^\mathcal{T}$ that contains an r-transition from A to A with weight 0.8 and an ε-transition from A to B with weight 0.6.

The weight of any word r^n in this automaton is easily computed to be 0.6. For any gfp-model $\mathcal{I}$ of $\mathcal{T}$ with underlying primitive interpretation $\mathcal{J}$, Lemma 35 shows that $A^\mathcal{I}(x)$ is determined by the smallest value of the form $0.6 \Rightarrow (\forall r^n.B)^{\mathcal{J}}(x)$ for any $n \geq 0$. If we have $(\forall r^n.B)^{\mathcal{J}}(x) \geq 0.6$ for all $n \geq 0$, i.e. if all paths from x to some y via r-connections in $\mathcal{J}$ satisfy the property that either the minimum degree of the r-edges on this path is $\leq B^{\mathcal{J}}(y)$, or else $B^{\mathcal{J}}(y) \geq 0.6$, then we know that $A^{\mathcal{I}}(x) = 1$. Otherwise, the value of $A^{\mathcal{I}}(x)$ is the infimum over the values of all such r-paths in $\mathcal{J}$.

Using the characterization of Lemma 35, it is possible to decide gfp-subsumptions by comparing the behavior of WWA, as described next.

Lemma 37. *Given $A, B \in \mathsf{N}_\mathsf{C}^\mathcal{T}$ and $p \in \mathfrak{C}$, A is gfp-subsumed by B to degree p w.r.t. $\mathcal{T}$ iff for all $C \in \mathsf{N}_\mathsf{P}^\mathcal{T}$ and $w \in \mathsf{N}_\mathsf{R}^*$ it holds that* $\min(p, \|\mathbf{A}_{B,C}^\mathcal{T}\|(w)) \leq \|\mathbf{A}_{A,C}^\mathcal{T}\|(w)$.

This means that, in order to decide gfp-subsumption between A and B, we have to compare the paths in $\mathbf{A}_{A,C}^\mathcal{T}$ and $\mathbf{A}_{B,C}^\mathcal{T}$ accepting the same words. From another perspective, we have to consider the weights of all matching terms $\forall w.C$ in the expanded definitions of A and B according to $\mathcal{T}$, and see whether they support the desired subsumption degree p.

Since $\mathsf{wt}_\mathcal{T}$ has finite support and takes only values from $\mathcal{V}_\mathcal{T}$, $\min(p, \|\mathbf{A}_{B,C}^\mathcal{T}\|(w)) > \|\mathbf{A}_{A,C}^\mathcal{T}\|(w)$ holds iff $\min(p', \|\mathbf{A}_{B,C}^\mathcal{T}\|(w)) > \|\mathbf{A}_{A,C}^\mathcal{T}\|(w)$, where p' is the smallest element of $\mathcal{V}_\mathcal{T}$ such that $p' \geq p$. This means that it suffices to be able to check gfp-subsumptions for the values in $\mathcal{V}_\mathcal{T}$. Rather than doing it with the WWA $\mathbf{A}_{B,C}^\mathcal{T}$ and $\mathbf{A}_{A,C}^\mathcal{T}$ directly, we will perform inclusion tests between polynomially many *unweighted* automata, which intuitively correspond to the cuts of the weighted automaton at different degrees.

Definition 38 (automata $\mathbf{A}_{\geq p}$)**.** Given a WWA $\mathbf{A} = (\Sigma, Q, q_0, \mathsf{wt}, q_f)$ and a value $p \in \mathfrak{C}$, the WA $\mathbf{A}_{\geq p} = (\Sigma, Q, q_0, \Delta_p, q_f)$ is given by the transition relation $\Delta_p := \{(q, w, q') \in Q \times \Sigma^* \times Q \mid \mathsf{wt}(q, w, q') \geq p\}$.

The language of this automaton has the obvious relation to the behavior of the original WWA $\mathbf{A}$:

$$L(\mathbf{A}_{\geq p}) = \{w \in \Sigma^* \mid \|\mathbf{A}\|(w) \geq p\}.$$

Using Lemma 37, we obtain the following characterization of gfp-subsumption in terms of language inclusions between WA.

Lemma 39. *Let $A, B \in \mathsf{N}_\mathsf{C}^\mathcal{T}$ and $p \in \mathcal{V}_\mathcal{T}$. Then A is subsumed by B to degree p w.r.t. $\mathcal{T}$ iff for all $C \in \mathsf{N}_\mathsf{P}^\mathcal{T}$ and $p' \in \mathcal{V}_\mathcal{T}$ with $p' \leq p$ it holds that $L((\mathbf{A}_{B,C}^\mathcal{T})_{\geq p'}) \subseteq L((\mathbf{A}_{A,C}^\mathcal{T})_{\geq p'})$.*

A direct consequence of this lemma is that gfp-subsumption between concept names in $\mathfrak{C}$-$\mathcal{FL}_0$ remains in the same complexity class as for classical $\mathcal{FL}_0$ (see Proposition 7).

Theorem 40. *Deciding gfp-subsumption between concept names in $\mathfrak{C}$-$\mathcal{FL}_0$ is* PSPACE*-complete.*

As in the case of $\mathfrak{C}$-$\mathfrak{N}\mathcal{AL}$, to compute the *best* gfp-subsumption degree between A and B, it suffices to check above inclusions for increasing values $p \in \mathcal{V}_\mathcal{T}$. The largest p for which these checks succeed is the requested degree. Thus, this degree can also be computed using only polynomial space.

6. Related Work

The results presented here are special cases of the more general constructions presented before in [14–16]. We describe first how far they can be extended.

The Hintikka automata for finite chains (Section 4.1) can be used even for arbitrary finite *lattices* L of truth degrees and with arbitrary t-norms instead of only the infimum function of the lattice. Moreover, the definitions can be extended to cover much more expressive DLs such as L-$\mathcal{ALCI}$ [42], L-$\mathcal{SHI}$ [16], and even L-$\mathcal{SHOI}$ [43]. Borrowing an automata-based technique originally developed for classical DLs from [13], it is possible to show lower complexity bounds for some of these fuzzy DLs, if the TBox is restricted to be *acyclic*; i.e., if the TBox contains only concept definitions, and there is no cyclic dependency between defined concepts. In particular, it has been shown that all the studied decision problems are PSPACE-complete in the logics L-$\mathcal{ALCHI}$, L-$\mathcal{ALCHO}$, L-$\mathcal{SI}_\mathsf{c}$, and L-$\mathcal{SO}_\mathsf{c}$ [16, 43].[9]

The ordered Hintikka trees (Section 4.2) can be adapted to deal with the implication constructor and even the involutive negation interpreted by the function $x \mapsto 1 - x$ [14]. Moreover, we believe that the techniques of [14] and [16] can be combined to derive tight complexity results for more expressive Gödel DLs and also under the restriction to acyclic TBoxes.

In the case of $\mathfrak{C}$-$\mathcal{FL}_0$, if the TBox is acyclic, then the automata $(\mathbf{A}^{\mathcal{T}}_{B,C})_{\geq p}$ obtained through our construction in Section 5 are, in fact, acyclic too. It is well-known that deciding the inclusion between two acyclic automata is in coNP [27]. Hence, deciding subsumption between concept names w.r.t. acyclic TBoxes is also in coNP, once again matching the complexity of the classical case.

In the following, we give a short overview of other research on fuzzy description logics. The first papers on fuzzy DLs considered Zadeh semantics over $[0, 1]$ and presented tableaux algorithms for reasoning in fuzzy $\mathcal{ALC}$ with acyclic TBoxes [21, 44]. However, these algorithms are not complexity-optimal, yielding EXPSPACE as the best known upper bound, compared to the lower bound of PSPACE from classical $\mathcal{ALC}$. These algorithms were later generalized to deal with GCIs [45] and more expressive DLs such as $\mathcal{SHIN}$ [32], $\mathcal{ALCIQ}$ [46], and even $\mathcal{SROIQ}$ [47].

For fuzzy DLs using *arbitrary t-norms* over $[0, 1]$, the presence of GCIs in combination with a negation constructor often leads to undecidability [7–10]. On the other hand, sometimes these logics become trivial in the sense that fuzzy ontologies are not more expressive than classical ones [48]. In the less expressive DL $\mathcal{EL}$ with GCIs, the Gödel t-norm over $[0, 1]$ does not increase the complexity of reasoning, i.e. it remains in PTIME [49]. However, there are indications that this is not true for other t-norms [50].

When restricting to *acyclic TBoxes*, fuzzy DLs over $[0, 1]$ with arbitrary t-norms are still decidable, and several tableaux algorithms were developed for this setting [33, 51–54]. The idea of these algorithms is to construct an abstract representation of a model in which concrete values of concepts are encoded by variables ranging over $[0, 1]$. According to the chosen t-norm, the values of these variables are then restricted by polynomial inequations. In this way, a system of inequations is constructed that is in the worst case of exponential size and has a solution iff a model of the input TBox exists. Again, this yields upper bounds (either NEXPTIME or EXPSPACE) different from the only known lower bound (PSPACE). The precise complexity of reasoning in these logics remains open.

The first paper dealing with *finite lattices* of truth degrees in fuzzy DLs considered a generalized Zadeh semantics for L-$\mathcal{ALC}$ with acyclic TBoxes and proposed a generalization of a classical tableau algorithm [55]. This was later extended to the more expressive logic L-$\mathcal{SHIN}$ [56].

The most popular approach for reasoning with fuzzy DLs over *finite chains* is to reduce the fuzzy ontology to a classical one and then employ optimized decision procedures for the resulting classical reasoning problems. This technique even allows to reduce GCIs. The idea is to translate every concept name A into finitely many classical concept names $A_{\geq p}$, $p \in L$, with the intention that the interpretation of $A_{\geq p}$ collects all those individuals that belong to A with a membership degree at least p. To encode the order on L, one has to introduce a new GCI $A_{\geq p_2} \sqsubseteq A_{\geq p_1}$ for every adjacent pair (p_1, p_2) in the total order on L. The same is done for all role names. Complex concepts and axioms are then recursively translated into classical concepts and axioms that employ the new concept and role names. Under generalized Zadeh semantics, the reduction is polynomial [57], showing EXPTIME-completeness of reasoning in L-$\mathcal{ALCH}$ with GCIs. However, for chains using different t-norms, such a translation can cause an exponential blowup in the size of the ontology, yielding suboptimal complexity upper bounds. This idea was applied to fuzzy DLs ranging from L-$\mathcal{ALCH}$ [30] up to L-$\mathcal{SROIQ}$ [29, 31, 58]. It can also be applied to fuzzy DLs using Zadeh semantics over $[0, 1]$ since reasoning in such logics can be restricted to the finitely many values occurring in the input TBox (and their negations) [59].

[9]The subscript c denotes the use of *crisp* roles, i.e. that the interpretation of all roles is restricted to the classical truth values 0 and 1.

7. Conclusions

We have illustrated the utility of automata-based techniques for reasoning in fuzzy description logics and hinted at extensions to more expressive DLs. Logics of the types $\mathsf{G}\text{-}\mathcal{L}$ and $L\text{-}\mathcal{L}$ for a finite chain (or lattice) L are of particular interest since they allow to use GCIs in combination with expressive concept constructors. The only other fuzzy semantics for which this is true is the traditional Zadeh semantics [21, 47], which has some shortcomings with regards to the used implication function [4]. On the other hand, expressive t-norm based fuzzy description logics supporting GCIs are often either undecidable [7–10] or do not support actual fuzzy reasoning [10, 48].

The goal of this paper was to introduce the reader to the current automata-based approaches for reasoning in fuzzy DLs, which have been successfully used for proving tight complexity bounds. As such, we have presented the ideas behind the three main constructions used in the area; namely: building tree-shaped models directly through the runs of an automaton; building an abstraction of these models, when the model cannot be constructed; and characterizing the interpretation of a concept in terms of regular languages. We expect that the simple cases presented here will aid in the understanding of the ideas underlying these techniques and encourage their application to other scenarios.

We want to point out three gaps that remain open in the complexity analysis of fuzzy description logics. First, fuzzy extensions of inexpressive DLs like $\mathcal{EL}$, $\mathcal{FL}_0$, and *DL-Lite* using semantics other than (finite or infinite) Gödel chains have so far received little attention in the literature [6, 50, 60]. Second, the precise complexity of reasoning with acyclic TBoxes in fuzzy DLs using t-norms over $[0, 1]$ (except the Gödel t-norm) is currently unknown. The only known tableaux algorithms for these logics are not optimal w.r.t. complexity [33, 51–54]. The last gap concerns very expressive DLs (above $\mathcal{SHOI}$) with semantics based on finite lattices. For $\mathcal{SROIQ}$ over finite chains, only the reductions to classical DLs from [29, 31, 58] are known, which may incur an exponential blowup in the size of the ontology in the worst case.

Acknowledgements

This work was partially supported by the German Research Foundation (DFG) under grant BA 1122/17-1 (FuzzyDL) and within the Cluster of Excellence 'Center for Advancing Electronics Dresden' (cfAED). The results were developed while R. Peñaloza was affiliated with TU Dresden, and the Center for Advancing Electronics Dresden, Germany.

[1] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation, and Applications, 2nd Edition, Cambridge University Press, 2007.

[2] L. A. Zadeh, Fuzzy sets, Information and Control 8 (3) (1965) 338–353. doi:10.1016/S0019-9958(65)90241-X.

[3] P. Cintula, P. Hájek, C. Noguera (Eds.), Handbook of Mathematical Fuzzy Logic, Vol. 37–38 of Studies in Logic, College Publications, 2011.

[4] F. Bobillo, M. Delgado, J. Gómez-Romero, A crisp representation for fuzzy $\mathcal{SHOIN}$ with fuzzy nominals and general concept inclusions, in: P. C. G. da Costa, C. d'Amato, N. Fanizzi, K. B. Laskey, K. J. Laskey, T. Lukasiewicz, M. Nickles, M. Pool (Eds.), Uncertainty Reasoning for the Semantic Web I, Vol. 5327 of Lecture Notes in Artificial Intelligence, Springer-Verlag, 2008, pp. 174–188. doi:10.1007/978-3-540-89765-1_11.

[5] P. Hájek, Metamathematics of Fuzzy Logic (Trends in Logic), Springer-Verlag, 2001.

[6] S. Borgwardt, R. Peñaloza, Positive subsumption in fuzzy $\mathcal{EL}$ with general t-norms, in: F. Rossi (Ed.), Proc. of the 23rd Int. Joint Conf. on Artificial Intelligence (IJCAI'13), AAAI Press, 2013, pp. 789–795.
URL http://ijcai.org/papers13/Abstracts/123.html

[7] F. Baader, R. Peñaloza, On the undecidability of fuzzy description logics with GCIs and product t-norm, in: C. Tinelli, V. Sofronie-Stokkermans (Eds.), Proc. of the 8th Int. Symp. on Frontiers of Combining Systems (FroCoS'11), Vol. 6989 of Lecture Notes in Computer Science, Springer-Verlag, 2011, pp. 55–70. doi:10.1007/978-3-642-24364-6_5.

[8] S. Borgwardt, R. Peñaloza, Undecidability of fuzzy description logics, in: G. Brewka, T. Eiter, S. A. McIlraith (Eds.), Proc. of the 13th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'12), AAAI Press, 2012, pp. 232–242.
URL http://www.aaai.org/ocs/index.php/KR/KR12/paper/view/4387

[9] M. Cerami, U. Straccia, On the (un)decidability of fuzzy description logics under Łukasiewicz t-norm, Information Sciences 227 (2013) 1–21. doi:10.1016/j.ins.2012.11.019.

[10] S. Borgwardt, F. Distel, R. Peñaloza, The limits of decidability in fuzzy description logics with general concept inclusions, Artificial Intelligence 218 (2015) 23–55. doi:10.1016/j.artint.2014.09.001.

[11] F. Baader, Using automata theory for characterizing the semantics of terminological cycles, Annals of Mathematics and Artificial Intelligence 18 (2) (1996) 175–219. doi:10.1007/BF02127747.

[12] M. Y. Vardi, P. Wolper, Automata-theoretic techniques for modal logics of programs, Journal of Computer and System Sciences 32 (2) (1986) 183–221. doi:10.1016/0022-0000(86)90026-7.

[13] F. Baader, J. Hladik, R. Peñaloza, Automata can show PSPACE results for description logics, Information and Computation 206 (9-10) (2008) 1045–1056. doi:10.1016/j.ic.2008.03.006.

[14] S. Borgwardt, F. Distel, R. Peñaloza, Decidable Gödel description logics without the finitely-valued model property, in: C. Baral, G. De Giacomo, T. Eiter (Eds.), Proc. of the 14th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'14), AAAI Press, 2014, pp. 228–237.
URL http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/7803

[15] S. Borgwardt, J. A. Leyva Galano, R. Peñaloza, The fuzzy description logic G-$\mathcal{FL}_0$ with greatest fixed-point semantics, in: E. Fermé, J. Leite (Eds.), Proc. of the 14th Eur. Conf. on Logics in Artificial Intelligence (JELIA'14), Vol. 8761 of Lecture Notes in Artificial Intelligence, Springer-Verlag, 2014, pp. 62–76.

[16] S. Borgwardt, R. Peñaloza, The complexity of lattice-based fuzzy description logics, Journal on Data Semantics 2 (1) (2013) 1–19. doi:10.1007/s13740-012-0013-x.

[17] J. N. Mordeson, D. S. Malik, Fuzzy automata and languages: theory and applications, CRC Press, 2010.

[18] S. Borgwardt, R. Peñaloza, Description logics over lattices with multi-valued ontologies, in: T. Walsh (Ed.), Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI'11), AAAI Press, 2011, pp. 768–773. doi:10.5591/978-1-57735-516-8/IJCAI11-135.

[19] E. P. Klement, R. Mesiar, E. Pap, Triangular Norms, Trends in Logic, Studia Logica Library, Springer-Verlag, 2000.

[20] P. Hájek, Making fuzzy description logic more general, Fuzzy Sets and Systems 154 (1) (2005) 1–15. doi:10.1016/j.fss.2005.03.005.

[21] U. Straccia, Reasoning within fuzzy description logics, Journal of Artificial Intelligence Research 14 (2001) 137–166. doi:10.1613/jair.813.

[22] F. Bobillo, U. Straccia, *fuzzyDL*: An expressive fuzzy description logic reasoner, in: Proc. of the 2008 IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'08), IEEE Computer Society Press, 2008, pp. 923–930. doi:10.1109/FUZZY.2008.4630480.

[23] D. Kang, B. Xu, J. Lu, Y. Li, Reasoning for a fuzzy description logic with comparison expressions, in: B. Parsia, U. Sattler, D. Toman (Eds.), Proc. of the 2006 Int. Workshop on Description Logics (DL'06), Vol. 189 of CEUR Workshop Proceedings, 2006, pp. 111–118.
URL http://ceur-ws.org/Vol-189/submission_1.pdf

[24] S. Borgwardt, Fuzzy description logics with general concept inclusions, Ph.D. thesis, Technische Universität Dresden, Germany (2014).

[25] A. Tarski, A lattice-theoretical fixpoint theorem and its applications, Pacific Journal of Mathematics 5 (2) (1955) 285–309. doi:10.2140/pjm.1955.5.285.

[26] J. E. Hopcroft, R. Motwani, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, 2nd Edition, Addison-Wesley, 2003.

[27] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman & Co., New York, 1979.

[28] K. Schild, A correspondence theory for terminological logics: Preliminary report, in: J. Mylopoulos, R. Reiter (Eds.), Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91), Morgan Kaufmann, 1991, pp. 466–471.
URL http://ijcai.org/Past%20Proceedings/IJCAI-91-VOL1/PDF/072.pdf

[29] F. Bobillo, M. Delgado, J. Gómez-Romero, U. Straccia, Joining Gödel and Zadeh fuzzy logics in fuzzy description logics, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 20 (4) (2012) 475–508. doi:10.1142/S0218488512500249.

[30] F. Bobillo, U. Straccia, Finite fuzzy description logics and crisp representations, in: F. Bobillo, P. C. G. da Costa, C. d'Amato, N. Fanizzi, K. Laskey, K. Laskey, T. Lukasiewicz, M. Nickles, M. Pool (Eds.), Uncertainty Reasoning for the Semantic Web II, Vol. 7123 of Lecture Notes in Computer Science, Springer-Verlag, 2013, pp. 102–121. doi:10.1007/978-3-642-35975-0_6.

[31] F. Bobillo, M. Delgado, J. Gómez-Romero, U. Straccia, Fuzzy description logics under Gödel semantics, International Journal of Approximate Reasoning 50 (3) (2009) 494–514. doi:10.1016/j.ijar.2008.10.003.

[32] G. Stoilos, G. B. Stamou, J. Z. Pan, V. Tzouvaras, I. Horrocks, Reasoning with very expressive fuzzy description logics, Journal of Artificial Intelligence Research 30 (2007) 273–320. doi:10.1613/jair.2279.

[33] U. Straccia, F. Bobillo, Mixed integer programming, general concept inclusions and fuzzy description logics, Mathware & Soft Computing 14 (3) (2007) 247–259.
URL http://ic.ugr.es/Mathware/index.php/Mathware/article/view/21

[34] F. Baader, R. Peñaloza, Are fuzzy description logics with general concept inclusion axioms decidable?, in: Proc. of the 2011 IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'11), IEEE Computer Society Press, 2011, pp. 1735–1742. doi:10.1109/FUZZY.2011.6007520.

[35] M. Baaz, A. Ciabattoni, N. Preining, First-order satisfiability in gödel logics: An NP-complete fragment, Theoretical Computer Science 412 (47) (2011) 6612–6623. doi:10.1016/j.tcs.2011.07.015.

[36] D. Guller, On the satisfiability and validity problems in the propositional Gödel logic, Computational Intelligence 399 (2012) 211–227. doi:10.1007/978-3-642-27534-0_14.

[37] B. Hollunder, Consistency checking reduced to satisfiability of concepts in terminological systems, Annals of Mathematics and Artificial Intelligence 18 (2-4) (1996) 133–157. doi:10.1007/BF02127745.

[38] L. Khachiyan, A polynomial algorithm in linear programming, Soviet Math. Doklady 20 (1) (1979) 191–194.

[39] N. Karmarkar, A new polynomial-time algorithm for linear programming, Proc. of the 16th Annual ACM Symp. on Theory of Computing (STOC'84) (1984) 302–311doi:10.1145/800057.808695.

[40] L. A. Wolsey, Integer programming, Wiley-Interscience, New York, NY, USA, 1998.
[41] F. Baader, S. Brandt, C. Lutz, Pushing the $\mathcal{EL}$ envelope, in: L. P. Kaelbling, A. Saffiotti (Eds.), Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI'05), Professional Book Center, 2005, pp. 364–369.
URL http://www.ijcai.org/papers/0372.pdf
[42] S. Borgwardt, R. Peñaloza, Finite lattices do not make reasoning in $\mathcal{ALCI}$ harder, in: F. Bobillo, R. Carvalho, P. C. G. da Costa, C. d'Amato, N. Fanizzi, K. B. Laskey, K. J. Laskey, T. Lukasiewicz, T. Martin, M. Nickles, M. Pool (Eds.), Proc. of the 7th Int. Workshop on Uncertainty Reasoning for the Semantic Web (URSW'11), Vol. 778 of CEUR Workshop Proceedings, 2011, pp. 51–62.
URL http://ceur-ws.org/Vol-778/paper5.pdf
[43] S. Borgwardt, Fuzzy DLs over finite lattices with nominals, in: M. Bienvenu, M. Ortiz, R. Rosati, M. Šimkus (Eds.), Proc. of the 27th Int. Workshop on Description Logics (DL'14), Vol. 1193 of CEUR Workshop Proceedings, 2014, pp. 58–70.
URL http://ceur-ws.org/Vol-1193/paper_4.pdf
[44] C. B. Tresp, R. Molitor, A description logic for vague knowledge, in: H. Prade (Ed.), Proc. of the 13th Eur. Conf. on Artificial Intelligence (ECAI'98), John Wiley and Sons, 1998, pp. 361–365.
[45] G. Stoilos, U. Straccia, G. B. Stamou, J. Z. Pan, General concept inclusions in fuzzy description logics, in: G. Brewka, S. Coradeschi, A. Perini, P. Traverso (Eds.), Proc. of the 17th Eur. Conf. on Artificial Intelligence (ECAI'06), IOS Press, 2006, pp. 457–461.
URL http://ebooks.iospress.nl/publication/2731
[46] G. Stoilos, G. B. Stamou, S. D. Kollias, Reasoning with qualified cardinality restrictions in fuzzy description logics, in: Proc. of the 2008 IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'08), IEEE Computer Society Press, 2008, pp. 637–644. doi:10.1109/FUZZY.2008.4630436.
[47] G. Stoilos, G. B. Stamou, Reasoning with fuzzy extensions of OWL and OWL 2, Knowledge and Information Systems 40 (1) (2014) 205–242. doi:10.1007/s10115-013-0641-y.
[48] S. Borgwardt, F. Distel, R. Peñaloza, How fuzzy is my fuzzy description logic?, in: B. Gramlich, D. Miller, U. Sattler (Eds.), Proc. of the 6th Int. Joint Conf. on Automated Reasoning (IJCAR'12), Vol. 7364 of Lecture Notes in Artificial Intelligence, Springer-Verlag, 2012, pp. 82–96. doi:10.1007/978-3-642-31365-3_9.
[49] T. Mailis, G. Stoilos, N. Simou, G. B. Stamou, S. Kollias, Tractable reasoning with vague knowledge using fuzzy $\mathcal{EL}^{++}$, Journal of Intelligent Information Systems 39 (2) (2012) 399–440. doi:10.1007/s10844-012-0195-6.
[50] S. Borgwardt, M. Cerami, R. Peñaloza, Many-valued horn logic is hard, in: T. Lukasiewicz, R. Peñaloza, A.-Y. Turhan (Eds.), Proc. of the 1st Workshop on Logics for Reasoning About Preferences, Uncertainty, and Vagueness (PRUV'14), Vol. 1205 of CEUR Workshop Proceedings, 2014, pp. 52–58.
URL http://ceur-ws.org/Vol-1205/00010052.pdf
[51] F. Bobillo, U. Straccia, A fuzzy description logic with product t-norm, in: Proc. of the 2007 IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'07), IEEE Computer Society Press, 2007, pp. 1–6. doi:10.1109/FUZZY.2007.4295443.
[52] F. Bobillo, U. Straccia, Fuzzy description logics with general t-norms and datatypes, Fuzzy Sets and Systems 160 (23) (2009) 3382–3402. doi:10.1016/j.fss.2009.03.006.
[53] G. Stoilos, G. B. Stamou, A framework for reasoning with expressive continuous fuzzy description logics, in: B. C. Grau, I. Horrocks, B. Motik, U. Sattler (Eds.), Proc. of the 2009 Int. Workshop on Description Logics (DL'09), Vol. 477 of CEUR Workshop Proceedings, 2009.
URL http://ceur-ws.org/Vol-477/paper_5.pdf
[54] V. Haarslev, H.-I. Pai, N. Shiri, A formal framework for description logics with uncertainty, International Journal of Approximate Reasoning 50 (9) (2009) 1399–1415. doi:10.1016/j.ijar.2009.04.009.
[55] U. Straccia, Uncertainty in description logics: A lattice-based approach, in: Proc. of the 10th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'04), 2004, pp. 251–258.
[56] Y. Jiang, Y. Tang, J. Wang, P. Deng, S. Tang, Expressive fuzzy description logics over lattices, Knowledge-Based Systems 23 (2) (2010) 150–161. doi:10.1016/j.knosys.2009.11.002.
[57] U. Straccia, Description logics over lattices, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 14 (1) (2006) 1–16. doi:10.1142/S0218488506003807.
[58] F. Bobillo, U. Straccia, Reasoning with the finitely many-valued Łukasiewicz fuzzy description logic $\mathcal{SROIQ}$, Information Sciences 181 (2011) 758–778. doi:10.1016/j.ins.2010.10.020.
[59] U. Straccia, Transforming fuzzy description logics into classical description logics, in: J. J. Alferes, J. A. Leite (Eds.), Proc. of the 9th Eur. Conf. on Logics in Artificial Intelligence (JELIA'04), Vol. 3229 of Lecture Notes in Computer Science, Springer-Verlag, 2004, pp. 385–399. doi:10.1007/978-3-540-30227-8_33.
[60] S. Borgwardt, R. Peñaloza, About subsumption in fuzzy $\mathcal{EL}$, in: T. Eiter, B. Glimm, Y. Kazakov, M. Krötzsch (Eds.), Proc. of the 26th Int. Workshop on Description Logics (DL'13), Vol. 1014 of CEUR Workshop Proceedings, 2013.
URL http://ceur-ws.org/Vol-1014/paper_48.pdf