Fuzzy DLs over Finite Lattices with Nominals*

Stefan Borgwardt

Theoretical Computer Science, TU Dresden, Germany
stefborg@tcs.inf.tu-dresden.de

Abstract. The complexity of reasoning in fuzzy description logics (DLs) over a finite lattice L usually does not exceed that of the underlying classical DLs. This has recently been shown for the logics between L-$\mathfrak{I}\mathcal{ALC}$ and L-$\mathfrak{I}\mathcal{SCHI}$ using a combination of automata- and tableau-based techniques. In this paper, this approach is modified to deal with nominals and constants in L-$\mathfrak{I}\mathcal{SCHOI}$. Reasoning w.r.t. general TBoxes is ExpTime-complete, and PSpace-completeness is shown under the restriction to acyclic terminologies in two sublogics. The latter implies two previously unknown complexity results for the classical DLs $\mathcal{ALCHO}$ and $\mathcal{SO}$.

1 Introduction

Fuzzy extensions of DLs have first been studied in [28,32,34] to model concepts that do not have a precise meaning. Such concepts occur in many application domains. For example, a physician may base a diagnosis on the patient having a *high fever*, which is not clearly characterized even by the precise body temperature. The main idea behind fuzzy DLs is that concepts are not interpreted as sets, but rather as fuzzy sets, which assign a *membership degree* from $[0, 1]$ to each domain element. As a fuzzy concept, HighFever could assign degree 0.7 to a patient with a body temperature of 38 °C, and 0.9 when the body temperature is 39 °C.

The first fuzzy DLs were based on the so-called *Zadeh* semantics that is derived from fuzzy set theory [35]. Later, it was proposed [19] to consider fuzzy DLs from the point of view of Mathematical Fuzzy Logic [18] and *t-norm-based* semantics were introduced. A t-norm is a binary operator on $[0, 1]$ that determines how the conjunction of two fuzzy statements is evaluated. Unfortunately, it was shown that many t-norm-based fuzzy DLs allowing general TBoxes have undecidable consistency problems [3,12,16]. This can be avoided by either choosing a t-norm that allows the consistency problem to be trivially reduced to classical reasoning [10], restricting to acyclic TBoxes [5], or taking the truth values from a finite structure, usually a total order [7,8,29] or a lattice [11,13,24,30]. Recently, it was shown that the complexity of reasoning in fuzzy DLs over finite lattices with (generalized) t-norms often matches that of the underlying classical DLs [13,14].

* Partially supported by the DFG under grant BA 1122/17-1 and in the research training group 1763 (QuantLA)

In this paper, we analyze the complexity of fuzzy extensions of $\mathcal{SHOI}$ using a finite lattice L. In the classical case, deciding consistency of ontologies with general TBoxes is ExpTime-complete in all logics between $\mathcal{ALC}$ and $\mathcal{SHOI}$ [20,26], and we show that this also holds for L-$\mathfrak{I}\mathcal{SCHOI}$. The additional letters $\mathfrak{I}$ and $\mathcal{C}$ in the name of the logic denote the presence of the constructors for implication and involutive negation, respectively. This nomenclature was introduced to make the subtle differences between different fuzzy DLs more explicit [12,15]. As all fuzzy DLs considered in this paper have both $\mathfrak{I}$ and $\mathcal{C}$, it is safe to ignore these letters here and simply read L-$\mathcal{SHOI}$ instead of L-$\mathfrak{I}\mathcal{SCHOI}$.

Consistency remains ExpTime-complete in the classical DLs $\mathcal{ALCOI}$ and $\mathcal{SH}$ even w.r.t. the *empty* TBox [22,31]. However, when restricting to acyclic (or empty) TBoxes in $\mathcal{SI}$, it is only PSpace-complete [1,23]. Similar results have been shown before under finite lattice semantics in L-$\mathfrak{I}\mathcal{ALCHI}$ and L-$\mathfrak{I}\mathcal{SCI}_{\mathsf{c}}$ [13]. The latter restricts all roles to be *crisp*, i.e. they are allowed to take only the two classical truth values. Here, we extend these results to L-$\mathfrak{I}\mathcal{ALCHO}$ and L-$\mathfrak{I}\mathcal{SCO}_{\mathsf{c}}$, which also shows previously unknown complexity results for the classical DLs $\mathcal{ALCHO}$ and $\mathcal{SO}$ with acyclic TBoxes. Due to space constraints, all proofs can be found only in the accompanying technical report [9].

2 Preliminaries

We first introduce looping automata on infinite trees and several helpful notions from [1], which will be used later for our reasoning procedures. Afterwards, we briefly recall relevant definitions from lattice theory [17].

2.1 Looping Automata

We consider automata on the (*unlabeled*) infinite tree of fixed arity $k \in \mathbb{N}$, represented by the set K^* of its *nodes*, where K abbreviates $\{1, \dots k\}$. Here, ε represents the root node, and ui, $i \in K$, is the i-th successor of the node $u \in K^*$. A *path* in this tree is a sequence $u_1, \dots, u_m$ of nodes such that $u_1 = \varepsilon$ and, for every i, $1 \leq i \leq m-1$, u_{i+1} is a successor of u_i.

Definition 1 (looping automaton). *A* looping (tree) automaton *is a tuple* $\mathsf{A} = (Q, I, \Delta)$ *where Q is a finite set of* states*, $I \subseteq Q$ is a set of* initial states*, and $\Delta \subseteq Q^{k+1}$ is the* transition relation*. A* run *of* A *is a mapping $r \colon K^* \to Q$ such that $r(\varepsilon) \in I$ and $(r(u), r(u1), \dots, r(uk)) \in \Delta$ for every $u \in K^*$. The* emptiness problem *is to decide whether a given looping automaton has a run.*

The emptiness problem for such automata is decidable in polynomial time [33]. However, the automata we construct in Section 4 are exponential in the size of the input. In order to obtain PSpace decision procedures, we need to identify the length of the longest possible path in a run that does not repeat any states.

Definition 2 (invariant, blocking). *Let* $\mathsf{A} = (Q, I, \Delta)$ *be a looping automaton and $\hookleftarrow$ a binary relation over Q, called the* blocking relation*.* A *is* $\hookleftarrow$-invariant *if*

$(q_0, q_1, \dots, q_i, \dots, q_k) \in \Delta$ *and* $q_i \hookleftarrow q_i'$ *always imply* $(q_0, q_1, \dots, q_i', \dots, q_k) \in \Delta$. *If this is the case, then* A *is* m-blocking *for* $m \in \mathbb{N}$ *if in every path* $u_1, \dots, u_m$ *of length* m *in a run* r *of* A *there are two indices* $1 \leq i < j \leq m$ *with* $r(u_j) \hookleftarrow r(u_i)$.

The notion of blocking is similar to that used in tableau algorithms for DLs [4,23]. If q is blocked by its ancestor q' ($q \hookleftarrow q'$), then we do not need to consider the subtree below q since every transition involving q can be replaced by one using q' instead. Of course, every looping automaton is $=$-invariant and $|Q|$-blocking. However, as mentioned above the size of Q may already be exponential in some external parameter. To obtain m-blocking automata with m bounded polynomially in the size of the input, we can use a faithful family of functions to prune the transition relation.

Definition 3 (faithful). *Let* $\mathsf{A} = (Q, I, \Delta)$ *be a looping automaton. A family* $\mathfrak{f} = (f_q)_{q \in Q}$ *of functions* $f_q \colon Q \to Q$ *is called* faithful *(w.r.t.* A*) if*

- *for all* $(q, q_1, \dots, q_k) \in \Delta$, *we have* $(q, f_q(q_1), \dots, f_q(q_k)) \in \Delta$, *and*
- *for all* $(q_0, q_1, \dots, q_k) \in \Delta$, *we have* $(f_q(q_0), f_q(q_1), \dots, f_q(q_k)) \in \Delta$.

The subautomaton $\mathsf{A}^{\mathfrak{f}} := (Q, I, \Delta^{\mathfrak{f}})$ *induced by* $\mathfrak{f}$ *is defined by*

$$\Delta^{\mathfrak{f}} := \{(q, f_q(q_1), \dots, f_q(q_k)) \mid (q, q_1, \dots, q_k) \in \Delta\}.$$

The name *faithful* reflects the fact that the resulting subautomaton simulates all runs of A. The following connection between the two automata was shown in [1].

Proposition 4. *Let* A *be a looping automaton and* $\mathfrak{f}$ *be a faithful family of functions for* A. *Then* A *has a run iff* $\mathsf{A}^{\mathfrak{f}}$ *has a run.*

Together with the following additional assumptions, polynomial blocking allows us to test emptiness in polynomial space.

Definition 5 (PSpace on-the-fly construction). *Let* I *be a set of* inputs. *A construction that yields, for each* $\mathsf{i} \in \mathsf{I}$, *an* m_{i}*-blocking looping automaton* A_{i} *over* k_{i}*-ary trees is called a* PSpace on-the-fly construction *if there is a polynomial* P *such that, for every input* i *of size* n,

(i) $m_{\mathsf{i}} \leq P(n)$ *and* $k_{\mathsf{i}} \leq P(n)$,
(ii) the size of every state of A_{i} *is bounded by* $P(n)$, *and*
(iii) one can guess in time bounded by $P(n)$ *an initial state, and, given a state* q, *a transition* $(q, q_1, \dots, q_k)$ *of* A_{i}.

The following result is again taken from [1].

Proposition 6. *If the looping automata* A_{i} *are obtained by a* PSpace *on-the-fly construction, then emptiness of* A_{i} *can be decided in* PSpace *in the size of* i.

2.2 Residuated Lattices

A *lattice* is an algebraic structure $(L, \vee, \wedge)$ with two commutative, associative, and idempotent binary operators, called *supremum* ($\vee$) and *infimum* ($\wedge$), that satisfy $x \wedge (x \vee y) = x$ and $x \vee (x \wedge y) = x$ for all $x, y \in L$. The natural partial order on L is given by $x \leq y$ iff $x \wedge y = x$ for all $x, y \in L$. An *antichain* is a set $S \subseteq L$ of incomparable elements. The *width* of a lattice is the maximal cardinality of all its antichains. A lattice L is *complete* if suprema and infima of arbitrary subsets $S \subseteq L$ exist; these are denoted by $\bigvee_{x \in S} x$ and $\bigwedge_{x \in S} x$, respectively. It is *distributive* if $\wedge$ and $\vee$ distribute over each other, *finite* if L is finite, and *bounded* if it has a least element $\mathbf{0}$ and a greatest element $\mathbf{1}$. Every finite lattice is complete, and every complete lattice is bounded by $\mathbf{0} := \bigwedge_{x \in L} x$ and $\mathbf{1} := \bigvee_{x \in L} x$.

A *De Morgan lattice* is a distributive lattice L with a unary involutive operator $\sim$ on L satisfying the De Morgan laws ${\sim}(x \vee y) = {\sim}x \wedge {\sim}y$ and ${\sim}(x \wedge y) = {\sim}x \vee {\sim}y$ for all $x, y \in L$. Such an operator is always antitone and satisfies ${\sim}\mathbf{0} = \mathbf{1}$. A *t-norm* over a bounded lattice L is a commutative, associative, monotone binary operator $\otimes$ on L that has $\mathbf{1}$ as its unit. A *residuated lattice* is a bounded lattice L with a t-norm $\otimes$ and a *residuum* $\Rightarrow : L \times L \to L$ satisfying $(x \otimes y) \leq z$ iff $y \leq (x \Rightarrow z)$ for all $x, y, z \in L$. We always assume that $\otimes$ is *join-preserving*, that is, $x \otimes \left(\bigvee_{y \in S} y\right) = \bigvee_{y \in S}(x \otimes y)$ holds for all $x \in L$ and $S \subseteq L$. This is a natural assumption that corresponds to the left-continuity assumption for t-norms over the standard fuzzy interval $[0, 1]$ [18].

3 L-$\mathcal{ISCHOI}$

Since fuzzy DLs over infinite lattices easily become undecidable when dealing with GCIs [3,12,14,16], we now fix a *finite* residuated De Morgan lattice L. For the complexity analysis, we assume that L is given as a list of its elements and that all lattice operations are computable in polynomial time.[1]

The syntax of the fuzzy description logic L-$\mathcal{ISCHOI}$ is similar to that of classical $\mathcal{SHOI}$: complex roles and concepts are constructed from disjoint sets $\mathsf{N_C}$ of *concept names*, $\mathsf{N_R}$ of *role names*, and $\mathsf{N_I}$ of *individual names*.

Definition 7 (syntax). *The set* $\mathsf{N_R^-}$ *of* (complex) roles *is* $\{r, r^- \mid r \in \mathsf{N_R}\}$. *The set of* (complex) concepts *is constructed as follows:*

- *every concept name is a concept, and*
- *for concepts C, D, $r \in \mathsf{N_R^-}$, $a \in \mathsf{N_I}$, and $p \in L$, the following are also concepts: $\overline{p}$ (constant), $\{a\}$ (nominal), $\neg C$ (negation), $C \sqcap D$ (conjunction), $C \to D$ (implication), $\exists r.C$ (existential restriction), and $\forall r.C$ (value restriction).*

For a complex role s, the *inverse of* s (written $\overline{s}$) is s^- if $s \in \mathsf{N_R}$ and r if $s = r^-$.

[1] If instead the size of the input encoding of L is logarithmic in the cardinality of L, then all complexity results except Theorem 18 remain valid.

Definition 8 (semantics). *A* (fuzzy) interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ *consists of a non-empty* domain $\Delta^\mathcal{I}$ *and an* interpretation function $\cdot^\mathcal{I}$ *that assigns to every* $A \in \mathsf{N_C}$ *a fuzzy set* $A^\mathcal{I}\colon \Delta^\mathcal{I} \to L$*, to every* $r \in \mathsf{N_R}$ *a fuzzy binary relation* $r^\mathcal{I}\colon \Delta^\mathcal{I} \times \Delta^\mathcal{I} \to L$*, and to every* $a \in \mathsf{N_I}$ *a domain element* $a^\mathcal{I} \in \Delta^\mathcal{I}$*. This function is extended to complex roles and concepts as follows for all* $x, y \in \Delta^\mathcal{I}$*:*

- $(r^-)^\mathcal{I}(x,y) := r^\mathcal{I}(y,x)$*;*
- $\overline{p}^\mathcal{I}(x) := p$*;*
- $\{a\}^\mathcal{I}(x) := \mathbf{1}$ *if* $x = a^\mathcal{I}$*, and* $\{a\}^\mathcal{I}(x) := \mathbf{0}$ *otherwise;*
- $(\neg C)^\mathcal{I}(x) := {\sim}C^\mathcal{I}(x)$*;*
- $(C \sqcap D)^\mathcal{I}(x) := C^\mathcal{I}(x) \otimes D^\mathcal{I}(x)$*;*
- $(C \to D)^\mathcal{I}(x) := C^\mathcal{I}(x) \Rightarrow D^\mathcal{I}(x)$*;*
- $(\exists r.C)^\mathcal{I}(x) := \bigvee_{y \in \Delta^\mathcal{I}} r^\mathcal{I}(x,y) \otimes C^\mathcal{I}(y)$*; and*
- $(\forall r.C)^\mathcal{I}(x) := \bigwedge_{y \in \Delta^\mathcal{I}} r^\mathcal{I}(x,y) \Rightarrow C^\mathcal{I}(y)$*.*

One can express *fuzzy nominals* [6] of the form $\{p_1/a_1, \dots, p_n/a_n\}$ with $p_i \in L$ and $a_i \in \mathsf{N_I}$, $1 \leq i \leq n$, by $(\{a_1\} \sqcap \overline{p_1}) \sqcup \dots \sqcup (\{a_n\} \sqcap \overline{p_n})$, where $C \sqcup D$ abbreviates $\neg(\neg C \sqcap \neg D)$. Unlike in classical DLs, existential and value restrictions need not be dual to each other, i.e. in general we have $(\neg\exists r.C)^\mathcal{I} \neq (\forall r.\neg C)^\mathcal{I}$.

Definition 9 (ontology). *An* axiom α *is a* concept assertion $\langle a{:}C \bowtie p\rangle$*, a* concept definition $\langle A \doteq C \geq p\rangle$*, a* general concept inclusion (GCI) $\langle C \sqsubseteq D \geq p\rangle$*, a* role inclusion $r \sqsubseteq s$*, or a* transitivity axiom $\mathsf{trans}(r)$*, where* C, D *are concepts,* $r, s \in \mathsf{N_R^-}$*,* $a \in \mathsf{N_I}$*,* $A \in \mathsf{N_C}$*,* $p \in L$*, and* $\bowtie\, \in \{<, \leq, =, \geq, >\}$*. An interpretation* $\mathcal{I}$ satisfies α *if* $C^\mathcal{I}(a^\mathcal{I}) \bowtie p$*,* $(A^\mathcal{I}(x) \Rightarrow C^\mathcal{I}(x)) \otimes (C^\mathcal{I}(x) \Rightarrow A^\mathcal{I}(x)) \geq p$*,* $C^\mathcal{I}(x) \Rightarrow D^\mathcal{I}(x) \geq p$*,* $r^\mathcal{I}(x,y) \leq s^\mathcal{I}(x,y)$*, or* $r^\mathcal{I}(x,y) \otimes r^\mathcal{I}(y,z) \leq r^\mathcal{I}(x,z)$*, respectively, hold for all* $x, y, z \in \Delta^\mathcal{I}$*.*

An acyclic TBox *is a finite set* $\mathcal{T}$ *of concept definitions where every* $A \in \mathsf{N_C}$ *has at most one definition* $\langle A \doteq C \geq p\rangle$ *in* $\mathcal{T}$ *and the relation* $>_\mathcal{T}$ *on* $\mathsf{N_C}$ *is acyclic, where* $A >_\mathcal{T} B$ *iff* B *occurs in the definition of* A*. A* general TBox *is a finite set of GCIs, an* ABox *a finite set of concept assertions, and an* RBox *a finite set of role inclusions and transitivity axioms. An* ontology *is a triple* $(\mathcal{A}, \mathcal{T}, \mathcal{R})$ *consisting of an ABox* $\mathcal{A}$*, an (acyclic or general) TBox* $\mathcal{T}$*, and an RBox* $\mathcal{R}$*. An interpretation is a model of this ontology if it satisfies all its axioms.*

We denote by $\mathsf{N_I}(\mathcal{O})$ and $\mathsf{N_R}(\mathcal{O})$ the sets of individual names and role names, respectively, occurring in an ontology $\mathcal{O}$, and set $\mathsf{N_R^-}(\mathcal{O}) := \{r, r^- \mid r \in \mathsf{N_R}(\mathcal{O})\}$. As usual, for an ontology $\mathcal{O} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ we define the *role hierarchy* $\sqsubseteq_\mathcal{R}$ as the reflexive transitive closure of $\{(r,s) \in \mathsf{N_R^-}(\mathcal{O}) \mid r \sqsubseteq s \in \mathcal{R} \text{ or } \overline{r} \sqsubseteq \overline{s} \in \mathcal{R}\}$, and we call a role r *transitive* if either $\mathsf{trans}(r) \in \mathcal{R}$ or $\mathsf{trans}(\overline{r}) \in \mathcal{R}$.

We do not consider *role assertions* of the form $\langle (a,b){:}r \bowtie p\rangle$ since in the presence of nominals they can be simulated by concept assertions, e.g. $\langle a{:}\exists r.\{b\} \bowtie p\rangle$.

The reasoning problem we consider in this paper is *consistency*, i.e. deciding the existence of a model for a given ontology. Due to the expressivity of our assertions, other reasoning problems such as satisfiability, subsumption, and instance checking can be reduced to consistency in linear time. For example, a concept C is p*-subsumed* by D w.r.t. $\mathcal{O} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$, i.e. we have $C^\mathcal{I}(x) \Rightarrow D^\mathcal{I}(x) \geq p$ for all models $\mathcal{I}$ of $\mathcal{O}$ and all $x \in \Delta^\mathcal{I}$, iff $(\mathcal{A} \cup \{\langle a{:}C \to D < p\rangle\}, \mathcal{T}, \mathcal{R})$ is inconsistent, where a is a fresh individual name.

4 Deciding Consistency

Consistency in L-$\mathfrak{I}\mathcal{SCHOI}$ with general TBoxes is ExpTime-complete, matching the complexity of classical $\mathcal{SHOI}$ [20]. To show this, we adapt the automata-based procedures from [1,13] to this more expressive logic. The conditions for the role hierarchy, inverse roles, and transitive roles are similar to the tableaux rules from [23]. To deal with nominals, we employ *pre-completions* inspired by the approaches in [2,14,21]. In Section 5, we derive complexity results for consistency in the sublogics L-$\mathfrak{I}\mathcal{ALCHO}$ (without transitivity and inverse roles) and L-$\mathfrak{I}\mathcal{SCO}_{\mathsf{c}}$ (without role inclusions, inverse roles, and fuzzy roles) with acyclic TBoxes.

It was shown in [13] that over a finite lattice L every interpretation $\mathcal{I}$ is n-*witnessed*, where n is the width of the lattice. This means that for every concept C, $r \in \mathsf{N}_\mathsf{R}^-$, and $x \in \Delta^\mathcal{I}$ there are n *witnesses* $y_1, \dots, y_n \in \Delta^\mathcal{I}$ such that $(\exists r.C)^\mathcal{I}(x) = \bigvee_{i=1}^n r^\mathcal{I}(x, y_i) \otimes C^\mathcal{I}(y_i)$, and similarly for the value restrictions. For the sake of simplicity, we present the following reasoning procedure only for the case of $n = 1$, i.e. we assume that all interpretations are *(1-)witnessed*. It can be generalized to handle arbitrary n by easy adaptations of the following definitions, in particular the introduction of more than one witness in Definition 12.

We now consider an ontology $\mathcal{O} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ that we want to test for consistency. The main idea of the algorithm is to find an abstract representation of a tree-shaped model of $\mathcal{O}$, a so-called *Hintikka tree*. Every node of this tree consists of a *Hintikka function* that describes the values of all relevant concepts for one domain element of the model. Additionally, each Hintikka function stores the values of all role connections from the parent node. We define the set $\mathsf{sub}(\mathcal{O})$ to contain all nominals $\{a\}$ for individual names $a \in \mathsf{N}_\mathsf{I}(\mathcal{O})$, all subconcepts of concepts occurring in $\mathcal{O}$, and all $\exists s.C$ (and $\forall s.C$) for which $\exists r.C$ ($\forall r.C$) occurs in $\mathcal{O}$, $s \sqsubseteq_\mathcal{R} r$, and s is transitive.

Definition 10 (Hintikka function). *A* Hintikka function *for $\mathcal{O}$ is a partial function $H\colon \mathsf{sub}(\mathcal{O}) \cup \mathsf{N}_\mathsf{R}^-(\mathcal{O}) \to L$ satisfying the following conditions:*

- *$H(s)$ is defined for all $s \in \mathsf{N}_\mathsf{R}^-(\mathcal{O})$;*
- *if $H(\overline{p})$ is defined, then $H(\overline{p}) = p$;*
- *if $H(\{a\})$ is defined, then $H(\{a\}) \in \{\mathbf{0}, \mathbf{1}\}$;*
- *if $H(C \sqcap D)$ is defined, then $H(C)$ and $H(D)$ are also defined and it holds that $H(C \sqcap D) = H(C) \otimes H(D)$; and similarly for $\neg C$ and $C \to D$.*

This function is compatible *with*

- *an assertion $\langle a{:}C \bowtie \ell\rangle$ if, whenever $H(\{a\}) = \mathbf{1}$, then $H(C)$ is defined and $H(C) \bowtie \ell$.*
- *a concept definition $\langle A \doteq C \geq \ell\rangle$ if, whenever $H(A)$ is defined, then $H(C)$ is defined and $(H(A) \Rightarrow H(C)) \otimes (H(C) \Rightarrow H(A)) \geq \ell$.*
- *a GCI $\langle C \sqsubseteq D \geq \ell\rangle$ if $H(C)$ and $H(D)$ are defined and $H(C) \Rightarrow H(D) \geq \ell$.*
- *a role inclusion $r \sqsubseteq s$ if $H(r) \leq H(s)$.*
- *an ABox/TBox/RBox/ontology if it is compatible with all axioms in it.*

The support *of H is the set* $\mathsf{supp}(H)$ *of all* $C \in \mathsf{sub}(\mathcal{O})$ *for which* H *is defined, and* $\mathsf{Ind}(H)$ *is the set of all* $a \in \mathsf{N_I}(\mathcal{O})$ *for which* $H(\{a\}) = \mathbf{1}$.

To deal with nominals, our algorithm maintains a polynomial amount of global information about the named domain elements, called a *pre-completion*. Since one domain element can have several names, we first consider a partition of $\mathsf{N_I}(\mathcal{O})$ that specifies which names are interpreted by the same elements. The pre-completion further contains one Hintikka function for each named individual, and the values of all role connections between them.

Definition 11 (pre-completion). *A* pre-completion *for the ontology $\mathcal{O}$ is a triple $\mathfrak{P} = (\mathcal{P}, \mathcal{H}_\mathcal{P}, \mathcal{R}_\mathcal{P})$, where $\mathcal{P}$ is a partition of $\mathsf{N_I}(\mathcal{O})$, $\mathcal{H}_\mathcal{P} = (H_X)_{X\in\mathcal{P}}$ is a family of Hintikka functions for $\mathcal{O}$, and $\mathcal{R}_\mathcal{P} = (r_\mathcal{P})_{r\in\mathsf{N_R}(\mathcal{O})}$ is a family of fuzzy binary relations $r_\mathcal{P} \colon \mathcal{P}\times\mathcal{P} \to L$, such that, for all $X \in \mathcal{P}$, we have $\mathsf{Ind}(H_X) = X$ and H_X is compatible with $\mathcal{O}$. A Hintikka function H for $\mathcal{O}$ is* compatible *with $\mathfrak{P}$ if for all $a \in \mathsf{Ind}(H)$, we have $H|_{\mathsf{sub}(\mathcal{O})} = H_{[a]_\mathcal{P}}|_{\mathsf{sub}(\mathcal{O})}$.*

We further set $r^-_\mathcal{P}(X, Y) := r_\mathcal{P}(Y, X)$ for all $X, Y \in \mathcal{P}$ and $r \in \mathsf{N_R}(\mathcal{O})$.

The arity k of our Hintikka trees is the number of existential and value restrictions in $\mathsf{sub}(\mathcal{O})$. Each successor in the tree describes the witness for one restriction. For the following definition, we consider $K := \{1, \dots, k\}$ as before and fix a bijection $\varphi \colon \{C \mid C \in \mathsf{sub}(\mathcal{O})$ is of the form $\exists r.D$ or $\forall r.D\} \to K$.

Definition 12 (Hintikka condition). *The tuple $(H_0, H_1, \dots, H_k)$ of Hintikka functions for $\mathcal{O}$ satisfies the* Hintikka condition *if the following hold:*

a) For every existential restriction $\exists r.C \in \mathsf{sub}(\mathcal{O})$:
   - *If $\exists r.C \in \mathsf{supp}(H_0)$ and $i = \varphi(\exists r.C)$, then we have $C \in \mathsf{supp}(H_i)$ and $H_0(\exists r.C) = H_i(r) \otimes H_i(C)$.*
   - *If $\exists r.C \in \mathsf{supp}(H_0)$, then for all $i \in K$, we have $C \in \mathsf{supp}(H_i)$ and $H_0(\exists r.C) \geq H_i(r) \otimes H_i(C)$; moreover, for all transitive roles $s \sqsubseteq_\mathcal{R} r$, we have $\exists s.C \in \mathsf{supp}(H_i)$ and $H_0(\exists r.C) \geq H_i(s) \otimes H_i(\exists s.C)$.*
   - *For all $i \in K$ with $\exists r.C \in \mathsf{supp}(H_i)$, we have $C \in \mathsf{supp}(H_0)$ and $H_i(\exists r.C) \geq H_i(\overline{r}) \otimes H_0(C)$; moreover, for all transitive roles $s \sqsubseteq_\mathcal{R} \overline{r}$, we have $\exists s.C \in \mathsf{supp}(H_0)$ and $H_i(\exists r.C) \geq H_i(s) \otimes H_0(\exists s.C)$.*

b) For every value restriction $\forall r.C \in \mathsf{sub}(\mathcal{O})$:
   - *If $\forall r.C \in \mathsf{supp}(H_0)$ and $i = \varphi(\forall r.C)$, then we have $C \in \mathsf{supp}(H_i)$ and $H_0(\forall r.C) = H_i(r) \Rightarrow H_i(C)$.*
   - *If $\forall r.C \in \mathsf{supp}(H_0)$, then for all $i \in K$, we have $C \in \mathsf{supp}(H_i)$ and $H_0(\forall r.C) \leq H_i(r) \Rightarrow H_i(C)$; moreover, for all transitive roles $s \sqsubseteq_\mathcal{R} r$, we have $\forall s.C \in \mathsf{supp}(H_i)$ and $H_0(\forall r.C) \leq H_i(s) \Rightarrow H_i(\forall s.C)$.*
   - *For all $i \in K$ with $\forall r.C \in \mathsf{supp}(H_i)$, we have $C \in \mathsf{supp}(H_0)$ and $H_i(\forall r.C) \leq H_i(\overline{r}) \Rightarrow H_0(C)$; moreover, for all transitive roles $s \sqsubseteq_\mathcal{R} \overline{r}$, we have $\forall s.C \in \mathsf{supp}(H_0)$ and $H_i(\forall r.C) \leq H_i(s) \Rightarrow H_0(\forall s.C)$.*

c) For all $r \in \mathsf{N}^-_\mathsf{R}(\mathcal{O})$ and $i, j \in K$ such that $a \in \mathsf{Ind}(H_i)$, $b \in \mathsf{Ind}(H_j)$, and $[a]_\mathcal{P} = [b]_\mathcal{P}$, we have $H_i(r) = H_j(r)$.

d) For all $a \in \mathsf{Ind}(H_0)$, $r \in \mathsf{N}^-_\mathsf{R}(\mathcal{O})$, $i \in K$, and $b \in \mathsf{Ind}(H_i)$, it holds that $H_i(r) = r_\mathcal{P}([a]_\mathcal{P}, [b]_\mathcal{P})$.

Intuitively, Condition a) ensures that the designated successor satisfies the witnessing condition for $\exists r.C$, and that the other successors do not interfere; this includes the parent node, which is a $\overline{r}$-predecessor. Additionally, existential restrictions are transferred along transitive roles, as in the $\forall_+$-rule from [23]. Conditions c) and d) are concerned with the behavior of named successors; in particular, the values for the role connections between named individuals specified by the pre-completion should be respected.

Given a pre-completion $\mathfrak{P} = (\mathcal{P}, \mathcal{H}_\mathcal{P}, \mathcal{R}_\mathcal{P})$, a *Hintikka tree for $\mathcal{O}$ starting with H_X*, $X \in \mathcal{P}$, is a mapping $\mathbf{T}$ that assigns to each $u \in K^*$ a Hintikka function $\mathbf{T}(u)$ for $\mathcal{O}$ that is compatible with $\mathcal{T}$, $\mathcal{R}$, and $\mathfrak{P}$ such that $\mathbf{T}(\varepsilon) = H_X$ and every tuple $(\mathbf{T}(u), \mathbf{T}(u1), \dots, \mathbf{T}(uk))$ satisfies the Hintikka condition.

The proof of the following lemma can be found in [9].

Lemma 13. *$\mathcal{O}$ is consistent iff there exist a pre-completion $\mathfrak{P} = (\mathcal{P}, \mathcal{H}_\mathcal{P}, \mathcal{R}_\mathcal{P})$ for $\mathcal{O}$ and, for each $X \in \mathcal{P}$, a Hintikka tree for $\mathcal{O}$ starting with H_X.*

The *Hintikka automaton* for $\mathcal{O}$ and H_X is the LA $\mathsf{A}_{\mathcal{O},H_X} := (Q_\mathcal{O}, I_{H_X}, \Delta_\mathcal{O})$, where $Q_\mathcal{O}$ consists of all pairs (H, i) of Hintikka functions H for $\mathcal{O}$ that are compatible with $\mathcal{T}$, $\mathcal{R}$, and $\mathfrak{P}$ and indices $i \in K$, $I_{H_X} := \{(H_X, 1)\}$, and $\Delta_\mathcal{O}$ is the set of all tuples $((H_0, i_0), (H_1, 1), \dots, (H_k, k))$ such that $(H_0, \dots, H_k)$ satisfies the Hintikka condition. It is easy to see that the first components of the runs of $\mathsf{A}_{\mathcal{O},H_X}$ are exactly the Hintikka trees for $\mathcal{O}$ starting with H_X, and the second components simply store the index of the existential or value restriction for which the state acts as a witness. By Lemma 13, consistency of $\mathcal{O}$ is thus equivalent to the existence of a pre-completion and the non-emptiness of the Hintikka automata $\mathsf{A}_{\mathcal{O},H_X}$ for each equivalence class X.

Since the number of pre-completions is bounded exponentially in the size of the input ($\mathcal{O}$ and L) and each pre-completion is of size polynomial in the size of the input, we can enumerate all pre-completions in exponential time and for each of them check emptiness of the polynomially many automata $\mathsf{A}_{\mathcal{O},H_X}$. Since the size of these automata is exponential in the size of the input, by [33] we obtain the following complexity result. ExpTime-hardness holds already in $\mathcal{ALC}$ [26].

Theorem 14. *In L-$\mathfrak{I}\mathcal{SCHOI}$ over a finite residuated De Morgan lattice L, consistency w.r.t. general TBoxes is* ExpTime*-complete.*

5 Acyclic TBoxes

We now extend the previous complexity results for lattice-based fuzzy DLs with acyclic TBoxes [13,14] by showing that consistency in L-$\mathfrak{I}\mathcal{ALCHO}$ and L-$\mathfrak{I}\mathcal{SCO}_\mathsf{c}$ is PSpace-complete in this setting. Recall that in L-$\mathfrak{I}\mathcal{SCO}_\mathsf{c}$, roles must always be interpreted as *crisp* functions that only take the values $\mathbf{0}$ and $\mathbf{1}$. Due to the absence of inverse roles, in the following we can restrict all definitions to use $\mathsf{N_R}(\mathcal{O})$ instead of $\mathsf{N_R^-}(\mathcal{O})$, and we can remove Condition d) and the last items of Conditions a) and b) from Definition 12 [9].

Let now $\mathcal{O} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ be such that $\mathcal{T}$ is acyclic. We can guess a triple $\mathfrak{P} = (\mathcal{P}, \mathcal{H}_\mathcal{P}, \mathcal{R}_\mathcal{P})$ and verify the conditions of Definition 11 in (nondeterministic) polynomial space. Thus, if emptiness of the polynomially many Hintikka automata $\mathsf{A}_{\mathcal{O},H_X}$ could be decided in polynomial space, we would obtain a PSpace upper bound for consistency [25]. The idea is to modify the construction of $\mathsf{A}_{\mathcal{O},H_X}$ using a faithful family of functions to obtain a PSpace on-the-fly construction. As in [13], these automata already satisfy most of Definition 5, except the polynomial bound on the maximal length a path before (equality) blocking occurs. The faithful families of functions we use are very similar to those employed in [13] for L-$\mathfrak{I}\mathcal{ALCHI}$ and L-$\mathfrak{I}\mathcal{SCI}_\mathsf{c}$.

For the subsequent constructions to work, we need to change the notion of compatibility of a Hintikka function H with $\mathfrak{P}$ to a weaker variant: we only require that for every $a \in \mathsf{Ind}(H)$ and every $C \in \mathsf{sub}(\mathcal{O})$ for which $H(C)$ is defined, $H_{[a]_\mathcal{P}}(C)$ is also defined and $H(C) = H_{[a]_\mathcal{P}}(C)$. This new definition does not work in the presence of inverse roles. However, in L-$\mathfrak{I}\mathcal{ALCHO}$ and L-$\mathfrak{I}\mathcal{SCO}_\mathsf{c}$, all previous results remain valid [9].

We now present a faithful family of functions for the case that $\mathcal{O}$ is formulated in L-$\mathfrak{I}\mathcal{ALCHO}$. For this, we denote by $\mathsf{rd}_\mathcal{T}(C)$ the role depth of the unfolding of a concept C w.r.t. the acyclic TBox $\mathcal{T}$, by $\mathsf{rd}_\mathcal{T}(H)$ for a Hintikka function H the maximal $\mathsf{rd}_\mathcal{T}(C)$ of a concept $C \in \mathsf{supp}(H)$, and by $\mathsf{sub}^{\leq n}(\mathcal{O})$ the restriction of $\mathsf{sub}(\mathcal{O})$ to concepts C with $\mathsf{rd}_\mathcal{T}(C) \leq n$.

Definition 15 (family $\mathfrak{f}$). *We define $\mathfrak{f} = (f_q)_{q \in Q_\mathcal{O}}$ for all $q = (H, i) \in Q_\mathcal{O}$ with $n := \mathsf{rd}_\mathcal{T}(H)$ and all $q' = (H', i') \in Q_\mathcal{O}$ by $f_q(q') := (H'', i')$, where, for every $C \in \mathsf{sub}(\mathcal{O})$ and $r \in \mathsf{N_R}(\mathcal{O})$,*

$$H''(C) := \begin{cases} H'(C) & \textit{if } C \in \mathsf{sub}^{\leq n-1}(\mathcal{O}), \\ \textit{undefined} & \textit{otherwise;} \end{cases} \quad H''(r) := \begin{cases} H'(r) & \textit{if } \mathsf{supp}(H) \neq \emptyset, \\ \mathbf{0} & \textit{otherwise.} \end{cases}$$

For all $q, q' \in Q_\mathcal{O}$, we have that $f_q(q')$ is again a state of $\mathsf{A}_{\mathcal{O},H_X}$ (according to the new definition of compatibility with $\mathfrak{P}$). The idea is to reduce the maximal role depth of the Hintikka function in every transition of the automaton. This works since in the absence of transitive roles the Hintikka condition only constrains the values of C for restrictions $\exists r.C$ or $\forall r.C$ that appear in the parent node. Thus, (equality) blocking occurs after polynomially many transitions.

We show in [9] that $\mathfrak{f}$ is faithful w.r.t. $\mathsf{A}_{\mathcal{O},H_X}$ and the construction of the induced subautomaton $\mathsf{A}^{\mathfrak{f}}_{\mathcal{O},H_X}$ from L, $\mathcal{O}$, and H_X constitutes a PSpace on-the-fly construction. By Propositions 4 and 6, we obtain the desired complexity result; PSpace-hardness holds already in classical $\mathcal{ALC}$ w.r.t. the empty TBox [27].

Theorem 16. *In L-$\mathfrak{I}\mathcal{ALCHO}$ over a finite residuated De Morgan lattice L, consistency w.r.t. acyclic TBoxes is* PSpace*-complete.*

For L-$\mathfrak{I}\mathcal{SCO}_\mathsf{c}$, the construction is a little more involved. Since now the interpretations of roles are restricted to $\mathbf{0}$ and $\mathbf{1}$, all Hintikka functions H for $\mathcal{O}$ need to satisfy the additional condition that $H(r) \in \{\mathbf{0}, \mathbf{1}\}$ for all $r \in \mathsf{N_R}(\mathcal{O})$. We further denote by $\varphi_r(\mathcal{O})$ for $r \in \mathsf{N_R}(\mathcal{O})$ the set of all indices $i \in K$ such that $i = \varphi(C)$ for a concept C of the form $\exists r.D$ or $\forall r.D$. We then replace K in Definition 12 by $\varphi_r(\mathcal{O})$. The idea is that in the absence of role inclusions it suffices to consider one role for each successor. The resulting definition is closer to the Hintikka condition from [13]. Lemma 13 remains valid under these modifications [9].

Given a Hintikka function H for $\mathcal{O}$ and a role name r, we define the set

$$H|_r := \{C \in \mathsf{supp}(H) \mid C = \exists r.D \text{ or } C = \forall r.D\}.$$

Definition 17 (family $\mathfrak{g}$). *We define $\mathfrak{g} = (g_q)_{q \in Q_\mathcal{O}}$ for all $q = (H, i) \in Q_\mathcal{O}$ with $n := \mathsf{rd}_\mathcal{T}(H)$ and all $q' = (H', i') \in Q_\mathcal{O}$ and $r' \in \mathsf{N_R}(\mathcal{O})$ such that $i' \in \varphi_{r'}(\mathcal{O})$ by $g_q(q') := (H'', i')$, where, for all $C \in \mathsf{sub}(\mathcal{O})$ and $r \in \mathsf{N_R}(\mathcal{O})$:*

$$P := \begin{cases} \mathsf{sub}^{\leq n}(\mathcal{O}) \cap H'|_{r'} & \text{if } r' \text{ is transitive,} \\ \emptyset & \text{otherwise;} \end{cases}$$

$$H''(C) := \begin{cases} H'(C) & \text{if } C \in \mathsf{sub}^{\leq n-1}(\mathcal{O}) \cup P, \\ \text{undefined} & \text{otherwise;} \end{cases}$$

$$H''(r) := \begin{cases} H'(r) & \text{if } \mathsf{supp}(H) \neq \emptyset \text{ and } r = r', \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

Again, the resulting pair (H'', i') is an element of $Q_\mathcal{O}$. Here, we cannot always reduce the role depth of the Hintikka functions, but have to keep all existential and value restrictions that are transferred along a transitive role r' to ensure that the family $\mathfrak{g}$ is faithful w.r.t. $\mathsf{A}_{\mathcal{O},H_X}$.

To prove that the induced subautomaton $\mathsf{A}^{\mathfrak{g}}_{\mathcal{O},H_X}$ is obtained by a PSPACE on-the-fly construction, we have to employ a complicated blocking condition that extends the one used for L-$\mathfrak{ISCI}_\mathsf{c}$ in [13]. The main problem is to show that every path in a run of this automaton involving the same transitive role r can be blocked after polynomially many steps. The idea is that along such a path, only the values of the concepts in $H|_r$ and their direct subconcepts are relevant. Polynomial blocking follows from the facts that r is crisp and the Hintikka condition thus guarantees the values of all these concepts to behave monotonically along this path. For details, see the full proof in [9].

Propositions 4 and 6 and [27] now entail the following result.

Theorem 18. *In L-$\mathfrak{ISCO}_\mathsf{c}$ over a finite residuated De Morgan lattice L, consistency w.r.t. acyclic TBoxes is* PSPACE*-complete.*

As a side effect, we obtain new, albeit not surprising, complexity results for the underlying classical description logics.

Corollary 19. *In classical $\mathcal{ALCHO}$ and $\mathcal{SO}$, consistency w.r.t. acyclic TBoxes is* PSPACE*-complete.*

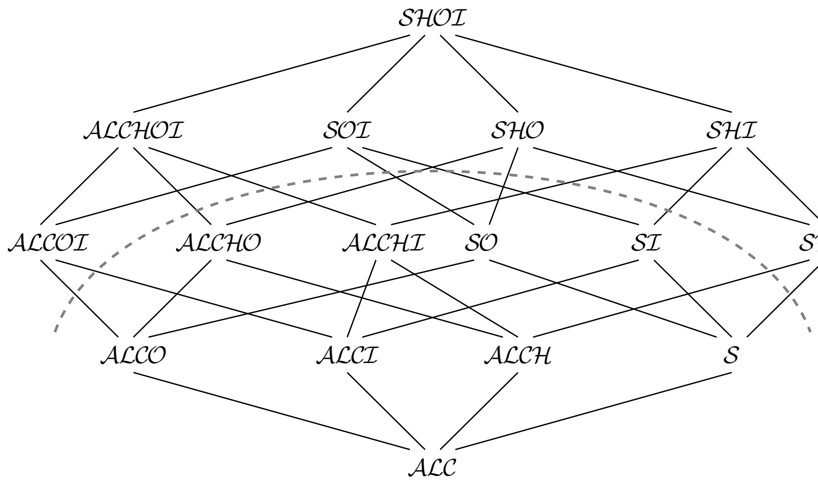

Fig. 1. The PSpace/ExpTime boundary in classical DLs with acyclic TBoxes

6 Conclusions

We have extended previous complexity results about fuzzy DLs with finite lattice semantics to cover nominals. This required extensive adaptations of the automata-based algorithm used for L-$\mathfrak{I}\mathcal{SCHI}$ and its sublogics in [13]. We employed pre-completions similar to those in [2,14,21] to show complexity results for ontology consistency. Due to the expressivity of our ABoxes, these easily transfer to other standard reasoning problems. In particular, we have shown that consistency in L-$\mathfrak{I}\mathcal{SCHOI}$ w.r.t. general TBoxes can be decided in ExpTime. This drops to PSpace when restricting to acyclic TBoxes in the sublogics L-$\mathfrak{I}\mathcal{ALCHO}$ and L-$\mathfrak{I}\mathcal{SCO}_{\mathsf{c}}$. To the best of our knowledge, only the sublogics $\mathcal{SI}$ [1,23] and $\mathcal{ALCHI}$ [13,14] of classical $\mathcal{SHOI}$ were known to have PSpace-complete reasoning problems w.r.t. acyclic TBoxes. On the other hand, in $\mathcal{ALCOI}$ and $\mathcal{SH}$ reasoning is already ExpTime-hard without any TBox [22,31]. The present results for $\mathcal{ALCHO}$ and $\mathcal{SO}$ thus complete the picture about reasoning w.r.t. acyclic TBoxes in the logics between $\mathcal{ALC}$ and $\mathcal{SHOI}$ (see Figure 1).

It would be interesting to extend the presented results to deal with fuzzy role inclusions ($\langle r \sqsubseteq s \geq p\rangle$) or cardinality restrictions ($\geq n\, r.C$), although it is not clear how to define the semantics of the latter in a setting where already a simple existential restriction may entail the existence of $n > 1$ witnessing role successors. We also plan to extend the automata-based algorithm for the fuzzy DL G-$\mathfrak{I}\mathcal{ALC}$ based on the so-called Gödel t-norm over the truth degrees from $[0, 1]$ to more expressive logics using the ideas presented here and in [13,14].

Acknowledgments

The author is indebted to Rafael Peñaloza for many discussions on the topics of (fuzzy) DLs in general and automata-based reasoning procedures in particular.

References

1. Baader, F., Hladik, J., Peñaloza, R.: Automata can show PSPACE results for description logics. Inform. Comput. 206(9-10), 1045–1056 (2008)
2. Baader, F., Lutz, C., Miličić, M., Sattler, U., Wolter, F.: Integrating description logics and action formalisms for reasoning about web services. LTCS-Report 05-02, Chair for Automata Theory, TU Dresden, Germany (2005), see http://lat.inf.tu-dresden.de/research/reports.html
3. Baader, F., Peñaloza, R.: On the undecidability of fuzzy description logics with GCIs and product t-norm. In: Proc. FroCoS'11, LNCS, vol. 6989, pp. 55–70. Springer (2011)
4. Baader, F., Sattler, U.: An overview of tableau algorithms for description logics. Studia Logica 69(1), 5–40 (2001)
5. Bobillo, F., Bou, F., Straccia, U.: On the failure of the finite model property in some fuzzy description logics. Fuzzy Set. Syst. 172(1), 1–12 (2011)
6. Bobillo, F., Delgado, M., Gómez-Romero, J.: A crisp representation for fuzzy $\mathcal{SHOIN}$ with fuzzy nominals and general concept inclusions. In: Uncertainty Reasoning for the Semantic Web I. LNAI, vol. 5327, pp. 174–188. Springer (2008)
7. Bobillo, F., Delgado, M., Gómez-Romero, J., Straccia, U.: Joining Gödel and Zadeh fuzzy logics in fuzzy description logics. Int. J. Uncertain. Fuzz. 20(4), 475–508 (2012)
8. Bobillo, F., Straccia, U.: Finite fuzzy description logics and crisp representations. In: Uncertainty Reasoning for the Semantic Web II, LNCS, vol. 7123, pp. 102–121. Springer (2013)
9. Borgwardt, S.: The complexity of fuzzy description logics over finite lattices with nominals. LTCS-Report 14-02, Chair for Automata Theory, TU Dresden, Germany (2014), see http://lat.inf.tu-dresden.de/research/reports.html.
10. Borgwardt, S., Distel, F., Peñaloza, R.: How fuzzy is my fuzzy description logic? In: Proc. IJCAR'12. LNAI, vol. 7364, pp. 82–96. Springer (2012)
11. Borgwardt, S., Peñaloza, R.: Description logics over lattices with multi-valued ontologies. In: Proc. IJCAI'11. pp. 768–773. AAAI Press (2011)
12. Borgwardt, S., Peñaloza, R.: Undecidability of fuzzy description logics. In: Proc. KR'12. pp. 232–242. AAAI Press (2012)
13. Borgwardt, S., Peñaloza, R.: The complexity of lattice-based fuzzy description logics. J. Data Semant. 2(1), 1–19 (2013)
14. Borgwardt, S., Peñaloza, R.: Consistency reasoning in lattice-based fuzzy description logics. Int. J. Approx. Reason. (2014), in press.
15. Cerami, M., Garcia-Cerdaña, À., Esteva, F.: From classical description logic to n-graded fuzzy description logic. In: Proc. FUZZ-IEEE'10. pp. 1–8. IEEE Press (2010)
16. Cerami, M., Straccia, U.: On the (un)decidability of fuzzy description logics under Łukasiewicz t-norm. Inform. Sciences 227, 1–21 (2013)
17. Galatos, N., Jipsen, P., Kowalski, T., Ono, H.: Residuated Lattices: An Algebraic Glimpse at Substructural Logics, Studies in Logic and the Foundations of Mathematics, vol. 151. Elsevier (2007)
18. Hájek, P.: Metamathematics of Fuzzy Logic (Trends in Logic). Springer (2001)
19. Hájek, P.: Making fuzzy description logic more general. Fuzzy Set. Syst. 154(1), 1–15 (2005)
20. Hladik, J.: To and Fro Between Tableaus and Automata for Description Logics. Ph.D. thesis, Technische Universität Dresden, Germany (2007)

21. Hollunder, B.: Consistency checking reduced to satisfiability of concepts in terminological systems. Ann. Math. Artif. Intell. 18(2-4), 133–157 (1996)
22. Horrocks, I.: Optimising Tableaux Decision Procedures for Description Logics. Ph.D. thesis, University of Manchester, UK (1997)
23. Horrocks, I., Sattler, U., Tobies, S.: Practical reasoning for very expressive description logics. L. J. IGPL 8(3), 239–263 (2000)
24. Jiang, Y., Tang, Y., Wang, J., Deng, P., Tang, S.: Expressive fuzzy description logics over lattices. Knowl.-Based Syst. 23(2), 150–161 (2010)
25. Savitch, W.J.: Relationships between nondeterministic and deterministic tape complexities. J. Comput. Syst. Sci. 4(2), 177–192 (1970)
26. Schild, K.: A correspondence theory for terminological logics: Preliminary report. In: Proc. IJCAI'91. pp. 466–471. Morgan Kaufmann (1991)
27. Schmidt-Schauß, M., Smolka, G.: Attributive concept descriptions with complements. Artif. Intell. 48(1), 1–26 (1991)
28. Straccia, U.: A fuzzy description logic. In: Proc. AAAI'98. pp. 594–599 (1998)
29. Straccia, U.: Transforming fuzzy description logics into classical description logics. In: Proc. JELIA'04. LNCS, vol. 3229, pp. 385–399. Springer (2004)
30. Straccia, U.: Uncertainty in description logics: A lattice-based approach. In: Proc. IPMU'04. pp. 251–258 (2004)
31. Tobies, S.: The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. J. Artif. Intell. Res. 12, 199–217 (2000)
32. Tresp, C.B., Molitor, R.: A description logic for vague knowledge. In: Proc. ECAI'98. pp. 361–365. John Wiley and Sons (1998)
33. Vardi, M.Y., Wolper, P.: Automata-theoretic techniques for modal logics of programs. J. Comput. Syst. Sci. 32(2), 183–221 (1986)
34. Yen, J.: Generalizing term subsumption languages to fuzzy logic. In: Proc. IJCAI'91. pp. 472–477. Morgan Kaufmann (1991)
35. Zadeh, L.A.: Fuzzy sets. Inform. Control 8(3), 338–353 (1965)