


Finite Lattices Do Not Make Reasoning in \mathcal{ALCOI} Harder

Stefan Borgwardt¹ and Rafael Peñaloza^{1,2}

¹ Theoretical Computer Science, TU Dresden, Dresden, Germany
{stefborg,penaloza}@tcs.inf.tu-dresden.de

² Center for Advancing Electronics Dresden, Dresden, Germany

Abstract. We consider the fuzzy description logic \mathcal{ALCOI} with semantics based on a finite residuated De Morgan lattice. We show that reasoning in this logic is EXPTIME-complete w.r.t. general TBoxes. In the sublogics \mathcal{ALCT} and \mathcal{ALCO} , it is PSPACE-complete w.r.t. acyclic TBoxes. This matches the known complexity bounds for reasoning in classical description logics between \mathcal{ALC} and \mathcal{ALCOI} .

1 Introduction

OWL 2, the current standard ontology language for the semantic web, is a syntactic variant of the crisp description logic (DL) $\mathcal{SROIQ}(\mathcal{D})$. The knowledge of an application domain can be formalized in such an ontology, and then reasoning problems such as ontology consistency, concept satisfiability, and concept subsumption can be used to infer new knowledge. As all crisp logics, this language is not well suited for expressing vague or imprecise notions that can be found in numerous domains. For instance, in the biomedical areas it is common to encounter concepts, such as `HighTemperature` or `Large`, that cannot be precisely represented using a classical logic.

Fuzzy extensions of DLs have been studied for the last two decades, and the literature on the topic is very extensive (see the surveys [27, 34]). Most of those approaches are based on the very simple *Zadeh semantics*, where the conjunction of two statements is interpreted as the minimum of their truth values; these values range over the interval $[0, 1]$ of rational numbers. The last decade has seen a shift towards more general semantics for treating vagueness, motivated by the development of mathematical fuzzy logic [23]. On the one hand, the use of continuous t-norms as the underlying interpretation function for conjunction was proposed in [24]. On the other hand, [33] considers incomparable truth degrees, which are structured into a lattice. However, this latter work still restricts to Zadeh-like semantics.

Most of the work on fuzzy DLs since then has focused on t-norm-based semantics over the unit interval; yet, even in those cases, ontologies are usually restricted to be unfoldable or acyclic [7–9]. Indeed, it has been shown that

Partially supported by the DFG under grant BA 1122/17-1, in the research training group 1763 (QuantLA), and in the Cluster of Excellence ‘cfAED’.

general concept inclusion axioms (GCIs) can cause undecidability even in fuzzy extensions of the basic DL \mathcal{ALC} [4, 5, 15, 16, 20]. In order to allow the expressivity of GCIs within the knowledge base, while retaining decidability, it is necessary to restrict the expressivity of the logic in other ways. This has motivated the study of fuzzy DLs over finitely-valued semantics [19]. The notion of t-norm can be rephrased in terms of finite lattices to preserve the relationship with mathematical fuzzy logic.

If one considers the Lukasiewicz t-norm over finitely many values, then reasoning is decidable even for very expressive DLs, as shown in [10] through a reduction to crisp reasoning. When restricted to \mathcal{ALC} without terminological axioms, concept satisfiability is PSPACE-complete as in the crisp case [18].¹ In the presence of general TBoxes, this problem becomes EXPTIME-complete [13, 14], again matching the complexity of the crisp case, even if arbitrary (finite) lattices and t-norms are allowed. However, the complexity of subsumption of concepts was left as an open problem, as the standard reduction used in crisp DLs does not work with general t-norm semantics.

In [11, 12, 17], matching complexity bounds were shown for other reasoning tasks and for logics up to \mathcal{SHI} , which extends \mathcal{ALC} with transitive and inverse roles, and allows for role inclusion axioms. More precisely, it was shown that all standard reasoning tasks are EXPTIME-complete in lattice-valued \mathcal{SHI} w.r.t. general TBoxes. If restricted to acyclic TBoxes, then the complexity reduces to PSPACE in \mathcal{ALCHI} ; the same holds for \mathcal{SI} under a restriction on the interpretation of the roles. For \mathcal{SH} , reasoning is EXPTIME-complete, even if the TBox is empty.

In this paper we complement those complexity results by showing that finite lattices do not affect the complexity of reasoning even if nominals are allowed, and provide tight complexity bounds for the fuzzy logic $L\text{-}\mathcal{ALCOI}$ over a finite lattice L . More precisely, we show that in this logic concept satisfiability is EXPTIME-complete w.r.t. general TBoxes and acyclic TBoxes. It was shown in [11, 17] that the restriction to acyclic TBoxes and the sublogic $L\text{-}\mathcal{ALCI}$ leads to a PSPACE-complete satisfiability problem. We show here that this is also the case in $L\text{-}\mathcal{ALCO}$. Moreover, the same complexity bounds hold for deciding subsumption between concepts. These results are in accordance with the complexity of reasoning in the classical DLs underlying these logics.

2 Preliminaries

We first recall some results about automata on infinite trees from [2] that will allow us to obtain tight upper bounds for our reasoning problems. Afterwards, we briefly introduce residuated lattices, which will be used for the semantics of our logic. For a more comprehensive view on residuated lattices, in particular in connection with mathematical fuzzy logic, we refer the reader to [21–23].

¹ The paper [18] considers the fuzzy modal logic \mathcal{K} , which can be seen as a syntactic variant of fuzzy \mathcal{ALC} with only one role.

2.1 Looping Automata on Infinite Trees

To obtain upper bounds for the complexity of reasoning in $L\text{-}\mathcal{ALCOI}$, we describe in Sect. 4 a reduction to the emptiness problem of looping automata on infinite trees. Such automata receive as input the unlabeled infinite k -ary tree for a fixed $k \in \mathbb{N}$. The *nodes* of this tree are represented by words in K^* , where $K := \{1, \dots, k\}$: the empty word ε represents the root node, and ui represents the i -th successor of the node u . An *ancestor* of a node $u \in K^*$ is a node $u' \in K^*$ for which there exists a $u'' \in K^*$ such that $u = u'u''$. A *path* is a sequence v_1, \dots, v_m of nodes such that $v_1 = \varepsilon$ and each v_{i+1} is a successor of v_i for every $i, 1 \leq i < m$.

Definition 1 (looping automaton). A looping (tree) automaton is a tuple $A = (Q, I, \Delta)$ 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 $r: K^* \rightarrow Q$ assigning states to each node of K^* such that $r(\varepsilon) \in I$ and for every $u \in K^*$, $(r(u), r(u1), \dots, r(uk)) \in \Delta$. The emptiness problem for looping automata is to decide whether a given looping automaton has a run.

The emptiness of looping automata can be decided in polynomial time using a bottom-up approach that finds all states that can appear in a run [37]. Alternatively, one can use a top-down approach, which relies on the fact that if there is a run, then there is also a periodic run. To speed up this search, the period should be as short as possible. This motivates the notion of *blocking automata*.

Definition 2 (m -blocking). Let $A = (Q, I, \Delta)$ be a looping automaton. We say that A is m -blocking for $m \in \mathbb{N}$ if every path v_1, \dots, v_m of length m in a run r of A contains two nodes v_i and v_j ($i < j$) such that $r(v_i) = r(v_j)$.

Clearly, every looping automaton is m -blocking for all $m > |Q|$. However, the main interest in blocking automata arises when one can find a smaller bound on m . One way to reduce this is through a so-called *faithful* family of functions.

Definition 3 (faithful). Let $A = (Q, I, \Delta)$ be a looping automaton. The family of functions $f_q: Q \rightarrow Q$, $q \in Q$, is faithful w.r.t. A if for all $q, q_0, q_1, \dots, q_k \in Q$,

- if $(q, q_1, \dots, q_k) \in \Delta$, then $(q, f_q(q_1), \dots, f_q(q_k)) \in \Delta$; and
- if $(q_0, q_1, \dots, q_k) \in \Delta$, then $(f_q(q_0), f_q(q_1), \dots, f_q(q_k)) \in \Delta$.

The subautomaton $A^S = (Q, I, \Delta^S)$ of A induced by this family has the transition relation $\Delta^S = \{(q, f_q(q_1), \dots, f_q(q_k)) \mid (q, q_1, \dots, q_k) \in \Delta\}$.

Lemma 4 [2]. Let A be a looping automaton and A^S its subautomaton induced by a faithful family of functions. Then A has a run iff A^S has a run.

The construction in Sect. 4 produces automata that are exponential in the size of the input. For such cases, it has been shown that if the automata are m -blocking for some m bounded polynomially in the size of the input (that is, logarithmically in the size of the automaton), then the emptiness test requires only polynomial space.

Definition 5 (PSPACE on-the-fly construction). Consider a set \mathcal{I} of inputs and a construction that yields, for every $i \in \mathcal{I}$, an m_i -blocking looping automaton $A_i = (Q_i, I_i, \Delta_i)$ on k_i -ary trees. This construction is a PSPACE on-the-fly construction if there is a polynomial P such that, for every input i of size n ,

- (i) $m_i \leq P(n)$ and $k_i \leq P(n)$,
- (ii) every element of Q_i has size bounded by $P(n)$, and
- (iii) one can nondeterministically guess in time bounded by $P(n)$ an element of I_i , and, for a state $q \in Q_i$, a transition from Δ_i with first component q .

As hinted at by the name, these conditions guarantee the following complexity result for checking emptiness of the constructed automata.

Theorem 6 [2]. If the looping automata A_i are obtained from the inputs $i \in \mathcal{I}$ by a PSPACE on-the-fly construction, then the emptiness of A_i can be checked in space polynomial in the size of i .

In Sect. 5, we will use this theorem to give PSPACE upper bounds on the complexity of reasoning in sublogics of $L\text{-}\mathcal{ALCCOI}$.

2.2 Residuated Lattices

A lattice is an algebraic structure (L, \vee, \wedge) over a carrier set L with two idempotent, associative, and commutative binary operations *join* \vee and *meet* \wedge that satisfy the absorption laws $\ell_1 \vee (\ell_1 \wedge \ell_2) = \ell_1 = \ell_1 \wedge (\ell_1 \vee \ell_2)$ for all $\ell_1, \ell_2 \in L$. The order \leq on L is uniquely defined by $\ell_1 \leq \ell_2$ iff $\ell_1 \wedge \ell_2 = \ell_1$ for all $\ell_1, \ell_2 \in L$. A lattice L is *distributive* if \vee and \wedge distribute over each other, *finite* if L is finite, and *bounded* if it has a *minimum* and a *maximum* element, denoted as $\mathbf{0}$ and $\mathbf{1}$, respectively. It is *complete* if joins and meets of arbitrary subsets $T \subseteq L$, denoted by $\bigvee_{t \in T} t$ and $\bigwedge_{t \in T} t$ respectively, exist. Every finite lattice is also bounded and complete. Whenever it is clear from the context, we simply use the carrier set L to represent the lattice (L, \vee, \wedge) .

A *De Morgan lattice* is a distributive lattice with an involutive and anti-monotonic unary operator \sim , called (*De Morgan*) *negation*, satisfying the De Morgan laws $\sim(\ell_1 \vee \ell_2) = \sim\ell_1 \wedge \sim\ell_2$ and $\sim(\ell_1 \wedge \ell_2) = \sim\ell_1 \vee \sim\ell_2$ for all $\ell_1, \ell_2 \in L$. For example, for every $n \in \mathbb{N}$, let $L_n = \{k/n \mid 0 \leq k \leq n\}$. Then (L_n, \max, \min) is a distributive lattice which, together with the negation $\sim\ell = 1 - \ell$, forms a De Morgan lattice.

An important notion in mathematical fuzzy logic is that of a triangular norm, or t-norm for short. We define this for arbitrary (bounded) lattices, although in the literature the term is usually only used in the context of the real interval $[0, 1]$ or finite chains [22, 23, 26].

Definition 7 (t-norm, residuum). Given a bounded lattice L , a (generalized) t-norm is an associative and commutative binary operator on L that is monotone w.r.t. the lattice order and has unit $\mathbf{1}$. A residuated lattice is a bounded lattice L extended with a t-norm \otimes and a binary operator \Rightarrow (called (generalized) residuum) such that, for all $\ell_1, \ell_2, \ell_3 \in L$, we have $\ell_1 \otimes \ell_2 \leq \ell_3$ iff $\ell_2 \leq \ell_1 \Rightarrow \ell_3$.

Notice that what we call a residuated lattice corresponds to a commutative, distributive, integral, zero-bounded FL-algebra from [22]. We chose to call it a residuated lattice to keep the relation with mathematical fuzzy logic explicit. A simple consequence of Definition 7 is that, for every $\ell_1, \ell_2 \in L$,

- $\mathbf{1} \Rightarrow \ell_1 = \ell_1$, and
- $\ell_1 \leq \ell_2$ iff $\ell_1 \Rightarrow \ell_2 = \mathbf{1}$.

For a t-norm \otimes over a complete lattice L , there is a binary operator \Rightarrow that satisfies the residuation property w.r.t. \otimes iff the t-norm is *join-preserving* [22], i.e. for all $\ell \in L$ and $T \subseteq L$ we have

$$\ell \otimes \left(\bigvee_{\ell' \in T} \ell' \right) = \bigvee_{\ell' \in T} (\ell \otimes \ell').$$

If \otimes also preserves arbitrary meets in the dual way, then \Rightarrow is uniquely determined as the function that satisfies, for all $\ell_1, \ell_2 \in L$,

$$\ell_1 \Rightarrow \ell_2 = \bigvee \{x \mid \ell_1 \otimes x \leq \ell_2\}.$$

Using this result, we often characterize a complete residuated lattice through its (join- and meet-preserving) t-norm, without explicitly mentioning its residuum. The t-norm and the De Morgan negation also uniquely determine the *t-conorm* $\ell_1 \oplus \ell_2 := \sim(\sim\ell_1 \otimes \sim\ell_2)$ and the *residual negation* $\ominus \ell := \ell \Rightarrow \mathbf{0}$.

For the rest of this paper, we fix a complete residuated De Morgan lattice L with De Morgan negation \sim and a join- and meet-preserving t-norm \otimes . The operators \Rightarrow, \oplus , and \ominus are then given by the above equations.

3 *L-ALCOI*

We now describe the fuzzy description logic *L-ALCOI*, whose semantics is based on the operators of L . It generalizes the classical DL *ALCOI* by using the elements of L as truth values, instead of just the Boolean *true* and *false*. The syntax of *L-ALCOI* is very similar to that of *ALCOI*. It is based on non-empty and pairwise disjoint sets \mathbf{N}_C , \mathbf{N}_R , and \mathbf{N}_I of *concept names*, *role names*, and *individual names*, respectively.

Definition 8 (syntax). A (complex) role is of the form r or r^- for $r \in \mathbf{N}_R$. (Complex) concepts are constructed from concept names using the constructors \top (top), $\{a\}$ (nominal for $a \in \mathbf{N}_I$), $\neg C$ (negation), $C \sqcap D$ (conjunction), $C \rightarrow D$ (implication), $\exists s.C$ (existential restriction for a complex role s), and $\forall s.C$ (value restriction).

For a complex role s , the *inverse* of s (denoted by \bar{s}) is s^- if $s \in \mathbf{N}_R$ and r if $s = r^-$. The main difference to the syntax of classical *ALCOI* is the explicit presence of the implication constructor.

The semantics of this logic is based on interpretation functions that map every concept C to a *fuzzy set* over the truth degrees from L , i.e. a function specifying the membership degree of every domain element to C .

Definition 9 (semantics). An interpretation is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set, called the domain of \mathcal{I} , and $\cdot^{\mathcal{I}}$ is an interpretation function that maps every concept name A to a function $A^{\mathcal{I}}: \Delta^{\mathcal{I}} \rightarrow L$, every role name r to a function $r^{\mathcal{I}}: \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow L$, and every individual name a to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. This function is extended to complex roles and concepts for all $x, y \in \Delta^{\mathcal{I}}$ as follows:

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

Note that we did not include the disjunction constructor, usually interpreted by the t-conorm, as it can be expressed using conjunction and negation. Likewise, the residual negation can be simulated by the implication, negation, and top. However, unlike in classical DLs, existential and universal quantifiers are not dual to each other, i.e. in general it does not hold that $(\neg \exists s.C)^{\mathcal{I}}(x) = (\forall s.\neg C)^{\mathcal{I}}(x)$.

The axioms of this logic also have an associated lattice value, which expresses the degree to which the restriction must be satisfied.

Definition 10 (axioms). An axiom is an assertion $\langle a:C \bowtie \ell \rangle$, a concept definition $\langle A \doteq C \geq \ell \rangle$, or a general concept inclusion (GCI) $\langle C \sqsubseteq D \geq \ell \rangle$, where $A \in \mathbf{N}_C$, $a \in \mathbf{N}_I$, $\ell \in L$, C, D are concepts, and $\bowtie \in \{<, \leq, =, \geq, >\}$. An ABox is a finite set of assertions. A general TBox is a finite set of GCIs. An acyclic TBox is a finite set of concept definitions such that every concept name occurs at most once on the left-hand side of an axiom, and there is no cyclic dependency between definitions. A TBox is either a general TBox or an acyclic TBox.² An ontology is a pair $\mathcal{O} = (\mathcal{A}, \mathcal{T})$ where \mathcal{A} is an ABox and \mathcal{T} is a TBox.

The interpretation \mathcal{I} satisfies (or is a model of)

- an assertion $\langle a:C \bowtie \ell \rangle$ if $C^{\mathcal{I}}(a^{\mathcal{I}}) \bowtie \ell$;
- a concept definition $\langle A \doteq C \geq \ell \rangle$ if for every element $x \in \Delta^{\mathcal{I}}$ it holds that $(A^{\mathcal{I}}(x) \Rightarrow C^{\mathcal{I}}(x)) \otimes (C^{\mathcal{I}}(x) \Rightarrow A^{\mathcal{I}}(x)) \geq \ell$;
- a GCI $\langle C \sqsubseteq D \geq \ell \rangle$ if for every $x \in \Delta^{\mathcal{I}}$ we have $C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x) \geq \ell$;
- an ABox, TBox, or ontology if it satisfies all axioms in it.

If \mathcal{T} is an acyclic TBox, then all concept names occurring on the left-hand side of some definition in \mathcal{T} are called *defined*, all others are called *primitive*. If \mathcal{T} is a general TBox, then all concept names appearing in it are *primitive*.

² We do not consider mixed TBoxes. We could allow axioms of the form $\langle A \sqsubseteq C \geq \ell \rangle$ in acyclic TBoxes, as long as they do not introduce cyclic dependencies. To avoid overloading the notation, we exclude this case.

Usually, ABoxes also contain *role assertions* of the form $\langle (a, b):r \bowtie \ell \rangle$, expressing that $r^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \bowtie \ell$ should hold. We do not consider such axioms here, since they can be simulated by the concept assertion $\langle a:\exists r.\{b\} \bowtie \ell \rangle$ (or $\langle b:\exists r^-. \{a\} \bowtie \ell \rangle$).

We emphasize that \mathcal{ALCCOI} is a special case of $L\text{-}\mathcal{ALCCOI}$, where the underlying lattice contains only the elements $\mathbf{0}$ and $\mathbf{1}$, which may be interpreted as *false* and *true*, respectively, and the t-norm is simply the classical conjunction. Accordingly, one can generalize the reasoning problems for \mathcal{ALCCOI} to lattices.

Definition 11 (reasoning). *Let C, D be concepts, \mathcal{O} an ontology, and $\ell \in L$.*

- \mathcal{O} is consistent if it has a model.
- C is ℓ -satisfiable w.r.t. \mathcal{O} if there is a model \mathcal{I} of \mathcal{O} and an element $x \in \Delta^{\mathcal{I}}$ such that $C^{\mathcal{I}}(x) \geq \ell$.
- C is ℓ -subsumed by D w.r.t. \mathcal{O} if every model of \mathcal{O} is also a model of $\langle C \sqsubseteq D \geq \ell \rangle$.
- The best satisfiability degree for C w.r.t. \mathcal{O} is the supremum of all $\ell' \in L$ such that C is ℓ' -satisfiable w.r.t. \mathcal{O} .
- The best subsumption degree of C and D w.r.t. \mathcal{O} is the supremum of all $\ell' \in L$ such that C is ℓ' -subsumed by D w.r.t. \mathcal{O} .

Unfortunately, consistency and satisfiability even in the smaller logic $L\text{-}\mathcal{ALC}$ are undecidable in general [1, 4, 5, 15, 16, 20]. Here we analyze the complexity of these problems under the assumption that L is *finite* and given as a list of its elements, and all lattice operations are computable in polynomial time in the size of their operands.

Observe that C is ℓ -satisfiable w.r.t. $(\mathcal{A}, \mathcal{T})$ iff $(\mathcal{A} \cup \{\langle a:C \geq \ell \rangle\}, \mathcal{T})$ is consistent, where a is a fresh individual name. Likewise, C is *not* ℓ -subsumed by D w.r.t. $(\mathcal{A}, \mathcal{T})$ iff $(\mathcal{A} \cup \{\langle a:C \rightarrow D < \ell \rangle\}, \mathcal{T})$ is consistent. To obtain the *best* degrees to which these inferences hold, one has to solve at most polynomially many consistency problems. For more details on these reductions, see e.g. [12]. We can thus focus on deciding consistency of ontologies to solve all other reasoning problems.

We show in Sect. 4 that the complexity of this problem is the same as for classical \mathcal{ALCCOI} : it is EXPTIME-complete w.r.t. both general and acyclic TBoxes [29, 30, 35, 36]. However, in the sublogics \mathcal{ALCCO} (without inverse roles) and \mathcal{ALCCI} (without nominals), consistency w.r.t. *acyclic TBoxes* is decidable in PSPACE [2, 3], and PSPACE-hard already in \mathcal{ALC} [31]. We show in Sect. 5 that these bounds also apply to $L\text{-}\mathcal{ALCCO}$ for arbitrary finite lattices L . The same holds for $L\text{-}\mathcal{ALCCI}$ [11, 17].

4 Consistency

We now show that consistency of $L\text{-}\mathcal{ALCCOI}$ ontologies is in EXPTIME. To achieve this, we adapt the approach from [2], where reasoning in classical DLs is reduced to the emptiness of an exponentially large looping automaton. To handle nominals and inverse roles correctly, we adapt ideas from [3, 25].

Recall that the semantics of the quantifiers requires the computation of a supremum or infimum of the membership degrees of a possibly infinite set of

elements of the domain. To obtain an effective decision procedure, reasoning is usually restricted to witnessed models [24].

Definition 12 (n -witnessed). *An interpretation \mathcal{I} is n -witnessed, $n \in \mathbb{N}$, if for every $x \in \Delta^{\mathcal{I}}$ and every concept of the form $\exists s.C$ there exist n elements $x_1, \dots, x_n \in \Delta^{\mathcal{I}}$ such that*

$$(\exists s.C)^{\mathcal{I}}(x) = \bigvee_{i=1}^n s^{\mathcal{I}}(x, x_i) \otimes C^{\mathcal{I}}(x_i),$$

and analogously for all value restrictions $\forall s.C$.

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

It was shown in [17] that every interpretation over the finite lattice L is n -witnessed for some n bounded by the cardinality of L . To simplify the description of the algorithm, in the following we consider only the case of $n = 1$. All constructions can easily be adapted for any other $n \in \mathbb{N}$.

Our algorithm for deciding consistency exploits the fact that an ontology \mathcal{O} has a model iff it has a well-structured *forest model*, consisting of interconnected tree-like structures rooted in the named individuals. We model them using so-called *Hintikka trees* that abstract from the complexity of a full model by only expressing the membership degrees for all relevant concepts. We construct automata that have exactly these Hintikka trees as their runs, and use the initial states to verify the restrictions imposed by the ABox. Reasoning is hence reduced to a polynomial guessing step and polynomially many emptiness tests for these automata.

In the following, we consider an ontology $\mathcal{O} = (\mathcal{A}, \mathcal{T})$ for which we want to decide consistency. We denote by $\text{sub}(\mathcal{O})$ the set of all subconcepts occurring in \mathcal{O} , and by $\text{Ind}(\mathcal{O})$ the set of all individual names occurring in \mathcal{O} . Similarly, the set $\text{Rol}(\mathcal{O})$ contains all role names used in \mathcal{O} , and $\text{Rol}^-(\mathcal{O})$ contains all complex roles occurring in \mathcal{O} . The nodes of the Hintikka trees are labeled with so-called Hintikka functions over the domain $\text{sub}(\mathcal{O}) \cup \{\rho\}$, 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.

Definition 13 (Hintikka function). *A Hintikka function for \mathcal{O} is a partial function $H: \text{sub}(\mathcal{O}) \cup \{\rho\} \rightarrow L$ satisfying the following conditions:*

- (i) $H(\rho)$ is defined;
- (ii) if $H(\top)$ is defined, then $H(\top) = \mathbf{1}$;
- (iii) if $H(\{a\})$ is defined, then $H(\{a\}) \in \{\mathbf{0}, \mathbf{1}\}$;
- (iv) if $H(\neg D)$ is defined, then $H(D)$ is defined and $H(\neg D) = \sim H(D)$;
- (v) 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
- (vi) if $H(C \rightarrow D)$ is defined, then $H(C)$ and $H(D)$ are also defined and it holds that $H(C \rightarrow D) = H(C) \Rightarrow H(D)$.

This function is compatible with

- an assertion $\langle a:C \bowtie \ell \rangle$ if $H(C)$ 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 $(H(A) \Rightarrow H(C)) \otimes (H(C) \Rightarrow H(A)) \geq \ell$;³
- a GCI $\langle C \sqsubseteq D \geq \ell \rangle$ if $H(C)$ and $H(D)$ are defined and $H(C) \Rightarrow H(D) \geq \ell$;
- an ABox/TBox if it is compatible with all axioms in it.

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

We denote by $H|_{\text{sub}(\mathcal{O})}$ the restriction of a Hintikka function H to $\text{sub}(\mathcal{O})$.

The first step of our decision procedure is to guess Hintikka functions for all named individuals. Since a domain element can have several names, we first need a partition \mathcal{P} of $\text{Ind}(\mathcal{O})$ that groups together all names referring to the same element. Given \mathcal{P} , we denote by $[a]_{\mathcal{P}}$ the class of \mathcal{P} that contains $a \in \text{Ind}(\mathcal{O})$. Then, we guess, for each $X \in \mathcal{P}$, one Hintikka function describing the behavior of the individual designated by the names in X . This is similar to the approach used in [3] to decide concept satisfiability in classical \mathcal{ALCOQ} with acyclic TBoxes. We additionally guess the values of the role connections between all named elements using fuzzy binary relations $r_{\mathcal{P}}$ on \mathcal{P} for every role name $r \in \text{Rol}(\mathcal{O})$.

Definition 14 (pre-completion). A pre-completion for the ontology \mathcal{O} is a triple $(\mathcal{P}, \mathcal{H}_{\mathcal{P}}, \mathcal{R}_{\mathcal{P}})$, where \mathcal{P} is a partition of $\text{Ind}(\mathcal{O})$, $\mathcal{H}_{\mathcal{P}} = (H_X)_{X \in \mathcal{P}}$ is a family of Hintikka functions for \mathcal{O} , and $\mathcal{R}_{\mathcal{P}} = (r_{\mathcal{P}})_{r \in \text{Rol}(\mathcal{O})}$ is a family of fuzzy binary relations $r_{\mathcal{P}}: \mathcal{P} \times \mathcal{P} \rightarrow L$ such that, for all $X \in \mathcal{P}$,

- $\text{Ind}(H_X) = X$;
- H_X is compatible with \mathcal{T} ; and
- H_X is compatible with $\mathcal{A}_X := \{\langle a:C \bowtie \ell \rangle \in \mathcal{A} \mid a \in X\}$.

A Hintikka function H for \mathcal{O} is compatible with this pre-completion if for all $a \in \text{Ind}(H)$, we have $H|_{\text{sub}(\mathcal{O})} = H_{[a]_{\mathcal{P}}}|_{\text{sub}(\mathcal{O})}$.

Each H_X is compatible with the pre-completion $(\mathcal{P}, \mathcal{H}_{\mathcal{P}}, \mathcal{R}_{\mathcal{P}})$ since, for every $a \in \text{Ind}(H_X) = X$ we have $[a]_{\mathcal{P}} = X$. We extend the family $\mathcal{R}_{\mathcal{P}}$ to complex roles by setting $r_{\overline{\mathcal{P}}}(X, Y) := r_{\mathcal{P}}(Y, X)$ for all $X, Y \in \mathcal{P}$.

Hintikka trees are k -ary trees labelled with Hintikka functions, where k is the number of existential and value restrictions in $\text{sub}(\mathcal{O})$. Intuitively, each successor acts as the witness for one of these restrictions. As in Sect. 2.1, we define $K := \{1, \dots, k\}$. Since we need to know which successor in the tree corresponds to which restriction, we fix an arbitrary bijection

$$\varphi: \{C \mid C \in \text{sub}(\mathcal{O}) \text{ is of the form } \exists s.D \text{ or } \forall s.D\} \rightarrow K,$$

and denote by $\varphi_s(\mathcal{O})$ for a role s the set of all indices $i \in K$ such that $i = \varphi(C)$ for a $C \in \text{sub}(\mathcal{O})$ of the form $\exists s.D$ or $\forall s.D$.

³ This method, called *lazy unfolding*, is only correct for acyclic TBoxes.

Definition 15 (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:*

- (i) *For every existential restriction $\exists s.C \in \text{sub}(\mathcal{O})$:*
 - *If $\exists s.C \in \text{supp}(H_0)$ and $i = \varphi(\exists s.C)$, then we have $C \in \text{supp}(H_i)$ and $H_0(\exists s.C) = H_i(\rho) \otimes H_i(C)$.*
 - *If $\exists s.C \in \text{supp}(H_0)$, then for all $i \in \varphi_s(\mathcal{O})$ we have $C \in \text{supp}(H_i)$ and $H_0(\exists s.C) \geq H_i(\rho) \otimes H_i(C)$.*
 - *For all $i \in \varphi_{\bar{s}}(\mathcal{O})$ with $\exists s.C \in \text{supp}(H_i)$, we have $C \in \text{supp}(H_0)$ and $H_i(\exists s.C) \geq H_i(\rho) \otimes H_0(C)$.*
- (ii) *For every universal restriction $\forall s.C \in \text{sub}(\mathcal{O})$:*
 - *If $\forall s.C \in \text{supp}(H_0)$ and $i = \varphi(\forall s.C)$, then we have $C \in \text{supp}(H_i)$ and $H_0(\forall s.C) = H_i(\rho) \Rightarrow H_i(C)$.*
 - *If $\forall s.C \in \text{supp}(H_0)$, then for all $i \in \varphi_s(\mathcal{O})$ we have $C \in \text{supp}(H_i)$ and $H_0(\forall s.C) \leq H_i(\rho) \Rightarrow H_i(C)$.*
 - *For all $i \in \varphi_{\bar{s}}(\mathcal{O})$ with $\forall s.C \in \text{supp}(H_i)$, we have $C \in \text{supp}(H_0)$ and $H_i(\forall s.C) \geq H_i(\rho) \Rightarrow H_0(C)$.*
- (iii) *For all $s \in \text{Rol}^-(\mathcal{O})$, $i, j \in \varphi_s(\mathcal{O})$, $a \in \text{Ind}(H_i)$, and $b \in \text{Ind}(H_j)$ with $[a]_{\mathcal{P}} = [b]_{\mathcal{P}}$, we have $H_i(\rho) = H_j(\rho)$.*
- (iv) *For all $a \in \text{Ind}(H_0)$, $s \in \text{Rol}^-(\mathcal{O})$, $i \in \varphi_s(\mathcal{O})$, and $b \in \text{Ind}(H_i)$, we have $H_i(\rho) = s_{\mathcal{P}}([a]_{\mathcal{P}}, [b]_{\mathcal{P}})$.*

Condition (i) makes sure that an existential restriction $\exists s.C$ is witnessed by its designated successor $\varphi(\exists s.C)$ and all other s -successors do not contradict the witness; this in particular includes possible \bar{s} -predecessors.⁴ Condition (ii) treats the universal restrictions analogously. Condition (iii) ensures that the value of a role connection to a named individual remains the same regardless of which index i is used to express it, and by Condition (iv) we enforce the previously guessed role connections between named elements.

Definition 16 (Hintikka tree). *Let $(\mathcal{P}, \mathcal{H}_{\mathcal{P}}, \mathcal{R}_{\mathcal{P}})$ be a pre-completion for \mathcal{O} and $X \in \mathcal{P}$. A Hintikka tree for \mathcal{O} starting with H_X is a mapping \mathbf{T} that assigns to each node $u \in K^*$ a Hintikka function for \mathcal{O} such that*

- (i) $\mathbf{T}(\varepsilon) = H_X$;
- (ii) *for every $u \in K^*$, $\mathbf{T}(u)$ is compatible with \mathcal{T} and the pre-completion; and*
- (iii) *for every $u \in K^*$, $(\mathbf{T}(u), \mathbf{T}(u1), \dots, \mathbf{T}(uk))$ satisfies the Hintikka condition.*

The definition of compatibility ensures that all axioms of \mathcal{T} are satisfied at every node of the Hintikka tree, while the Hintikka condition makes sure that the tree satisfies the witnessing conditions for all relevant quantified concepts.

The proof of the following theorem uses arguments similar to those in [2, 17]. The main difference to the classical case is the presence of successors witnessing the universal restrictions. We additionally have to deal here with the side condition of compatibility with the pre-completion.

⁴ There is at most one such predecessor, namely the parent node.

Lemma 17. \mathcal{O} is consistent iff there are a pre-completion $(\mathcal{P}, \mathcal{H}_{\mathcal{P}}, \mathcal{R}_{\mathcal{P}})$ for \mathcal{O} and, for each $X \in \mathcal{P}$, a Hintikka tree for \mathcal{O} starting with H_X .

Proof. Assume that a pre-completion and the required Hintikka trees \mathbf{T}_X for \mathcal{O} starting with H_X exist. We first remove irrelevant nodes in these Hintikka trees. A node $u \in K^*$ is *relevant* in \mathbf{T}_X if $\text{Ind}(\mathbf{T}_X(u')) = \emptyset$ for all (non-empty) ancestors $u' \in K^+$ of u . The idea is that if $a \in \text{Ind}(\mathbf{T}_X(u'))$, then by the compatibility with the pre-completion $\mathbf{T}_X(u')$ agrees with $H_{[a]_{\mathcal{P}}} = \mathbf{T}_{[a]_{\mathcal{P}}}(\varepsilon)$ on the values of all concepts in $\text{sub}(\mathcal{O})$, and thus $\mathbf{T}_X(u')$ can be replaced with $\mathbf{T}_{[a]_{\mathcal{P}}}(\varepsilon)$. The root nodes are always relevant since they are needed to represent the named individuals. We now define the interpretation \mathcal{I} with domain

$$\Delta^{\mathcal{I}} := \{(X, u) \in \mathcal{P} \times K^* \mid u \text{ is relevant in } \mathbf{T}_X\}.$$

We set $a^{\mathcal{I}} := ([a]_{\mathcal{P}}, \varepsilon)$ for all $a \in \text{Ind}(\mathcal{O})$. For $r \in \mathbf{N}_R$ and $(X, u), (Y, v) \in \Delta^{\mathcal{I}}$,

- $r^{\mathcal{I}}((X, u), (Y, v)) := \mathbf{T}_X(ui)(\rho)$ if there is an index $i \in \varphi_r(\mathcal{O})$ such that either
 - (i) $(Y, v) = (X, ui)$ or
 - (ii) $v = \varepsilon$ and $\text{Ind}(\mathbf{T}_X(ui)) \cap Y \neq \emptyset$;
- $r^{\mathcal{I}}((X, u), (Y, v)) := \mathbf{T}_Y(vi)(\rho)$ if there is an index $i \in \varphi_{r^-}(\mathcal{O})$ such that either
 - (i) $(X, u) = (Y, vi)$ or
 - (ii) $u = \varepsilon$ and $\text{Ind}(\mathbf{T}_Y(vi)) \cap X \neq \emptyset$; and
- $r^{\mathcal{I}}((X, u), (Y, v)) := \mathbf{0}$ otherwise.

To see that this is well-defined, consider the following three cases.

- If there are $i, j \in \varphi_r(\mathcal{O})$ such that $v = \varepsilon$ and both $\text{Ind}(\mathbf{T}_X(ui)) \cap Y$ and $\text{Ind}(\mathbf{T}_X(vj)) \cap Y$ are non-empty, then from Condition (iii) of Definition 15 we obtain $\mathbf{T}_X(ui)(\rho) = \mathbf{T}_X(vj)(\rho)$.
- For the dual case of $i, j \in \varphi_{r^-}(\mathcal{O})$ with $u = \varepsilon$, $\text{Ind}(\mathbf{T}_Y(vi)) \cap X \neq \emptyset$, and $\text{Ind}(\mathbf{T}_Y(vj)) \cap X \neq \emptyset$, we have $\mathbf{T}_Y(vi)(\rho) = \mathbf{T}_Y(vj)(\rho)$ by the same condition.
- If $u = v = \varepsilon$ and there are $i \in \varphi_r(\mathcal{O})$, $j \in \varphi_{r^-}(\mathcal{O})$, $a \in \text{Ind}(\mathbf{T}_X(i)) \cap Y$, and $b \in \text{Ind}(\mathbf{T}_Y(j)) \cap X$, then we have $Y = [a]_{\mathcal{P}}$ and $X = [b]_{\mathcal{P}}$. By Condition (iv) of Definition 15, this implies $\mathbf{T}_X(i)(\rho) = r_{\mathcal{P}}(X, Y) = r_{\overline{\mathcal{P}}}(Y, X) = \mathbf{T}_Y(j)(\rho)$.

We now define the interpretation of concept names. For a primitive concept name A , we simply set $A^{\mathcal{I}}(X, u) := \mathbf{T}_X(u)(A)$ for all $(X, u) \in \Delta^{\mathcal{I}}$. \mathcal{I} is extended to the defined concept names while showing the following claim:

$$\begin{aligned} &\text{for all } (X, u) \in \Delta^{\mathcal{I}} \text{ and all } C \in \text{sub}(\mathcal{O}) \text{ for which } \mathbf{T}_X(u)(C) \text{ is defined,} \\ &\text{we have } C^{\mathcal{I}}(X, u) = \mathbf{T}_X(u)(C). \end{aligned} \quad (1)$$

We prove this by induction on the (non-negative) *weight* $o(C)$, which is defined inductively as follows:

- $o(A) := o(\top) := o(\{a\}) := 0$ for every primitive concept name A and every $a \in \mathbf{N}_I$;
- $o(A) := o(C) + 1$ for every definition $\langle A \doteq C \geq \ell \rangle \in \mathcal{T}$;
- $o(\neg C) := o(C) + 1$;
- $o(C \sqcap D) := o(C \rightarrow D) := \max\{o(C), o(D)\} + 1$; and
- $o(\exists s.C) := o(\forall s.C) := o(C) + 1$.

This weight is well-defined for general and acyclic TBoxes.

For $C = \top$, Claim (1) follows immediately from Definition 13. For a primitive concept name A , it holds by the definition of $A^{\mathcal{I}}$ above.

If $\mathbf{T}_X(u)(\{a\})$ is defined for some $a \in \text{Ind}(\mathcal{O})$, then by Definition 13 this value is either $\mathbf{0}$ or $\mathbf{1}$. If it is $\mathbf{0}$, then we cannot have $\mathbf{T}_X(u) = H_{[a]_{\mathcal{P}}}$ by Definition 14. Thus, $a^{\mathcal{I}} = ([a]_{\mathcal{P}}, \varepsilon) \neq (X, u)$, and hence $\{a\}^{\mathcal{I}}(X, u) = \mathbf{0} = \mathbf{T}_X(u)(\{a\})$. Otherwise, we have $\mathbf{T}_X(u)(\{a\}) = \mathbf{1}$, i.e. $a \in \text{Ind}(\mathbf{T}_X(u))$. Since u is relevant in \mathbf{T}_X , we infer that $u = \varepsilon$. By Definition 14, we get $a \in \text{Ind}(\mathbf{T}_X(u)) = \text{Ind}(H_X) = X$, and thus $a^{\mathcal{I}} = ([a]_{\mathcal{P}}, \varepsilon) = (X, u)$. We conclude $\{a\}^{\mathcal{I}}(X, u) = \mathbf{1} = \mathbf{T}_X(u)(\{a\})$.

Consider now a defined concept name A with the definition $\langle A \doteq C \geq \ell \rangle \in \mathcal{T}$. If $\mathbf{T}_X(u)(A)$ is defined, then by the compatibility with \mathcal{T} the value $\mathbf{T}_X(u)(C)$ is also defined and $(\mathbf{T}_X(u)(A) \Rightarrow \mathbf{T}_X(u)(C)) \otimes (\mathbf{T}_X(u)(C) \Rightarrow \mathbf{T}_X(u)(A)) \geq \ell$. Since $o(C) < o(A)$, we get $C^{\mathcal{I}}(X, u) = \mathbf{T}_X(u)(C)$ by induction. Thus, we can define $A^{\mathcal{I}}(X, u) := \mathbf{T}_X(u)(A)$ to ensure that \mathcal{I} satisfies $\langle A \doteq C \geq \ell \rangle$ at (X, u) . Whenever $\mathbf{T}_X(u)(A)$ is undefined, we can set $A^{\mathcal{I}}(X, u) := C^{\mathcal{I}}(X, u)$ to satisfy this concept definition without violating the claim.

If $\mathbf{T}_X(u)(\neg C)$ is defined, then $\mathbf{T}_X(u)(C)$ is also defined. By induction, we obtain $(\neg C)^{\mathcal{I}}(X, u) = \sim C^{\mathcal{I}}(X, u) = \sim \mathbf{T}_X(u)(C) = \mathbf{T}_X(u)(\neg C)$. Similar arguments show Claim (1) for conjunctions and implications.

Assume now that $\ell := \mathbf{T}_X(u)(\exists s.C)$ is defined for a complex role s and a concept C , and let $i := \varphi(\exists s.C)$. We first prove the existence of a witness $(Y, v) \in \Delta^{\mathcal{I}}$ such that $s^{\mathcal{I}}((X, u), (Y, v)) \otimes C^{\mathcal{I}}(Y, v) = \ell$. By the Hintikka condition, we know that $\mathbf{T}_X(ui)(C)$ is defined and $\ell = \mathbf{T}_X(ui)(\rho) \otimes \mathbf{T}_X(ui)(C)$. Since u is relevant in \mathbf{T}_X , ui can only be irrelevant in \mathbf{T}_X if $\text{Ind}(\mathbf{T}_X(ui)) \neq \emptyset$. We make a case distinction on whether ui is relevant or not. (1) If there exists an $a \in \text{Ind}(\mathbf{T}_X(ui))$, then the compatibility of $\mathbf{T}_X(ui)$ with the pre-completion implies that $\mathbf{T}_{[a]_{\mathcal{P}}}(\varepsilon)(C) = H_{[a]_{\mathcal{P}}}(C) = \mathbf{T}_X(ui)(C)$ is defined. Since the root node ε is relevant in $\mathbf{T}_{[a]_{\mathcal{P}}}$, by induction we get $C^{\mathcal{I}}([a]_{\mathcal{P}}, \varepsilon) = \mathbf{T}_{[a]_{\mathcal{P}}}(\varepsilon)(C)$. Furthermore, by the definition of $s^{\mathcal{I}}$ we know that $s^{\mathcal{I}}((X, u), ([a]_{\mathcal{P}}, \varepsilon)) = \mathbf{T}_X(ui)(\rho)$, and thus we can choose the witness $(Y, v) := ([a]_{\mathcal{P}}, \varepsilon)$. (2) Otherwise, $\text{Ind}(\mathbf{T}_X(ui)) = \emptyset$ and $(X, ui) \in \Delta^{\mathcal{I}}$. By induction, we have $C^{\mathcal{I}}(X, ui) = \mathbf{T}_X(ui)(C)$, and from the definition of $s^{\mathcal{I}}$ we obtain $s^{\mathcal{I}}((X, u), (X, ui)) = \mathbf{T}_X(ui)(\rho)$, which allows us to choose $(Y, v) := (X, ui)$. It remains to show that $s^{\mathcal{I}}((X, u), (Z, w)) \otimes C^{\mathcal{I}}(Z, w) \leq \ell$ holds for all other $(Z, w) \in \Delta^{\mathcal{I}}$, which implies that $(\exists s.C)^{\mathcal{I}}(X, u) = \ell$, as desired. In the case that $s^{\mathcal{I}}((X, u), (Z, w)) = \mathbf{0}$, the claim is trivial. Otherwise, one of the following two alternatives must hold:

- There is an index $i \in \varphi_s(\mathcal{O})$ with $s^{\mathcal{I}}((X, u), (Z, w)) = \mathbf{T}_X(ui)(\rho)$ and
 - (i) $(Z, w) = (X, ui)$ or
 - (ii) $w = \varepsilon$ and $\text{Ind}(\mathbf{T}_X(ui)) \cap Z \neq \emptyset$. From the Hintikka condition we know that the value $\mathbf{T}_X(ui)(C)$ is defined and satisfies $\ell = \mathbf{T}_X(u)(\exists s.C) \geq \mathbf{T}_X(ui)(\rho) \otimes \mathbf{T}_X(ui)(C)$. It thus suffices to show that $C^{\mathcal{I}}(Z, w) = \mathbf{T}_X(ui)(C)$. In the first case, $\mathbf{T}_Z(w)(C) = \mathbf{T}_X(ui)(C)$ is defined, and thus induction yields that $C^{\mathcal{I}}(Z, w) = \mathbf{T}_Z(w)(C) = \mathbf{T}_X(ui)(C)$. In Case (ii), we know that $\mathbf{T}_Z(\varepsilon)(C) = H_Z(C) = \mathbf{T}_X(ui)(C)$ by the compatibility of $\mathbf{T}_X(ui)$ with the pre-completion. By induction we thus obtain $C^{\mathcal{I}}(Z, w) = C^{\mathcal{I}}(Z, \varepsilon) = \mathbf{T}_Z(\varepsilon)(C) = \mathbf{T}_X(ui)(C)$.

- There is an index $i \in \varphi_{\bar{s}}(\mathcal{O})$ with $s^{\mathcal{I}}((X, u), (Z, w)) = \mathbf{T}_Z(wi)(\rho)$ and (i') $(X, u) = (Z, wi)$ or (ii') $u = \varepsilon$ and $\text{Ind}(\mathbf{T}_Z(wi)) \cap X \neq \emptyset$. In Case (i'), we immediately get $\ell = \mathbf{T}_X(u)(\exists s.C) = \mathbf{T}_Z(wi)(\exists s.C)$. In the latter case, from the compatibility of $\mathbf{T}_Z(wi)$ with the pre-completion we also get that $\ell = \mathbf{T}_X(u)(\exists s.C) = \mathbf{T}_X(\varepsilon)(\exists s.C) = H_X(\exists s.C) = \mathbf{T}_Z(wi)(\exists s.C)$. Thus, in both cases the Hintikka condition yields that $\mathbf{T}_Z(w)(C)$ is defined and we have $\ell = \mathbf{T}_Z(wi)(\exists s.C) \geq \mathbf{T}_Z(wi)(\rho) \otimes \mathbf{T}_Z(w)(C)$. By induction, we obtain $\mathbf{T}_Z(w)(C) = C^{\mathcal{I}}(Z, w)$, which proves the claim.

Claim (1) can be shown for value restrictions using similar arguments.

We have thus defined an interpretation \mathcal{I} that satisfies all concept definitions in \mathcal{T} . In the case that \mathcal{T} is a general TBox, consider any GCI $\langle C \sqsubseteq D \geq \ell \rangle \in \mathcal{T}$ and $(X, u) \in \Delta^{\mathcal{I}}$. By the compatibility of $\mathbf{T}_X(u)$ with \mathcal{T} , we know that $\mathbf{T}_X(u)(C)$ and $\mathbf{T}_X(u)(D)$ are defined and $\mathbf{T}_X(u)(C) \Rightarrow \mathbf{T}_X(u)(D) \geq \ell$. By Claim (1), we thus have $C^{\mathcal{I}}(X, u) \Rightarrow D^{\mathcal{I}}(X, u) \geq \ell$, which shows that \mathcal{I} satisfies the GCI. Finally, consider an assertion $\langle a:C \bowtie \ell \rangle \in \mathcal{A}$. By the compatibility of $H_{[a]_{\mathcal{P}}}$ with $\mathcal{A}_{[a]_{\mathcal{P}}}$ (see Definition 14), we know that $H_{[a]_{\mathcal{P}}}(C)$ is defined and $H_{[a]_{\mathcal{P}}}(C) \bowtie \ell$. By Claim (1), we conclude $C^{\mathcal{I}}(a^{\mathcal{I}}) = C^{\mathcal{I}}([a]_{\mathcal{P}}, \varepsilon) = \mathbf{T}_{[a]_{\mathcal{P}}}(\varepsilon)(C) = H_{[a]_{\mathcal{P}}}(C) \bowtie \ell$; that is, \mathcal{I} satisfies the assertion.

Conversely, assume that there is a model \mathcal{I} of \mathcal{O} . We define a pre-completion $(\mathcal{P}, \mathcal{H}_{\mathcal{P}}, \mathcal{R}_{\mathcal{P}})$ for \mathcal{O} based on the partition $\mathcal{P} := \{\{b \in \text{Ind} \mid a^{\mathcal{I}} = b^{\mathcal{I}}\} \mid a \in \text{Ind}\}$. For every $r \in \text{Rol}(\mathcal{O})$ and $X, Y \in \mathcal{P}$, we set $r_{\mathcal{P}}(X, Y) := r^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}})$, where (a, b) is an arbitrary element of $X \times Y$. Similarly, we set $H_X(\rho) := \mathbf{0}$ and $H_X(C) := C^{\mathcal{I}}(a^{\mathcal{I}})$ for every $C \in \text{sub}(\mathcal{O})$ to define the family $\mathcal{H}_{\mathcal{P}} = (H_X)_{X \in \mathcal{P}}$. Since \mathcal{I} satisfies \mathcal{T} , this obviously defines Hintikka functions that are compatible with \mathcal{T} , and we also have $\text{Ind}(H_X) = X$ for every $X \in \mathcal{P}$. Furthermore, for every $\langle a:C \bowtie \ell \rangle \in \mathcal{A}$, we have $C^{\mathcal{I}}(a^{\mathcal{I}}) \bowtie \ell$, and thus $H_{[a]_{\mathcal{P}}}(C) \bowtie \ell$, which shows that what we have defined above is indeed a pre-completion for \mathcal{O} .

For a given $X \in \mathcal{P}$, we now define the Hintikka tree \mathbf{T}_X starting with H_X by inductively constructing a mapping $g_X: K^* \rightarrow \Delta^{\mathcal{I}}$ that specifies which elements of $\Delta^{\mathcal{I}}$ represent the nodes of \mathbf{T}_X and satisfies the following property:

$$\text{For all } u \in K^*, C \in \text{sub}(\mathcal{O}), s \in \text{Rol}^-(\mathcal{O}), \text{ and } i \in \varphi_s(\mathcal{O}), \text{ we have} \quad (2)$$

$$\mathbf{T}_X(u)(C) = C^{\mathcal{I}}(g_X(u)) \text{ and } \mathbf{T}_X(ui)(\rho) = s^{\mathcal{I}}(g_X(u), g_X(ui)).$$

This in particular ensures that all constructed Hintikka functions are compatible with \mathcal{T} and with the pre-completion.

We start the construction by setting $\mathbf{T}_X(\varepsilon) := H_X$ and $g_X(\varepsilon) := a^{\mathcal{I}}$, where a is an arbitrary element of X . Thus, \mathbf{T}_X starts with H_X and Claim (2) is satisfied at ε . Let now $u \in K^*$ be any node for which \mathbf{T}_X and g_X have already been defined while satisfying Claim (2), and consider any existential restriction $\exists s.C \in \text{sub}(\mathcal{O})$ and $i := \varphi(\exists s.C)$. Since \mathcal{I} is witnessed, there must be a $y \in \Delta^{\mathcal{I}}$ such that $(\exists s.C)^{\mathcal{I}}(g_X(u)) = s^{\mathcal{I}}(g_X(u), y) \otimes C^{\mathcal{I}}(y)$. We now set $g_X(ui) := y$, $\mathbf{T}_X(ui)(\rho) := s^{\mathcal{I}}(g_X(u), y)$, and $\mathbf{T}_X(ui)(C) := C^{\mathcal{I}}(y)$ for all $C \in \text{sub}(\mathcal{O})$ to satisfy Claim (2) at ui . Likewise, for any $\forall s.C \in \text{sub}(\mathcal{O})$ there must be a $y \in \Delta^{\mathcal{I}}$ with $(\forall s.C)^{\mathcal{I}}(g_X(u)) = s^{\mathcal{I}}(g_X(u), y) \Rightarrow C^{\mathcal{I}}(y)$, and we proceed as above to define \mathbf{T}_X and g_X at ui for $i := \varphi(\forall s.C)$.

We now show that every tuple $(\mathbf{T}_X(u), \mathbf{T}_X(u1), \dots, \mathbf{T}_X(uk))$ with $u \in K^*$ satisfies the Hintikka condition. The first point of Condition (i) from Definition 15 is obviously satisfied by the above construction. Consider now any $\exists s.C \in \text{sub}(\mathcal{O})$ and $i \in \varphi_s(\mathcal{O})$. By Claim (2) and the semantics of existential restrictions, we obtain

$$\begin{aligned} \mathbf{T}_X(u)(\exists s.C) &= (\exists s.C)^{\mathcal{I}}(g_X(u)) \\ &\geq s^{\mathcal{I}}(g_X(u), g_X(ui)) \otimes C^{\mathcal{I}}(g_X(ui)) \\ &= \mathbf{T}_X(ui)(\rho) \otimes \mathbf{T}_X(ui)(C). \end{aligned}$$

Similarly, for all $i \in \varphi_{\bar{s}}(\mathcal{O})$, we have

$$\begin{aligned} \mathbf{T}_X(ui)(\exists s.C) &= (\exists s.C)^{\mathcal{I}}(g_X(ui)) \\ &\geq s^{\mathcal{I}}(g_X(ui), g_X(u)) \otimes C^{\mathcal{I}}(g_X(u)) \\ &= \mathbf{T}_X(ui)(\rho) \otimes \mathbf{T}_X(u)(C). \end{aligned}$$

Condition (ii) can be shown by analogous arguments. For Condition (iii), let $u \in K^*$, $s \in \text{Rol}^-(\mathcal{O})$, $i, j \in \varphi_s(\mathcal{O})$, $a \in \text{Ind}(\mathbf{T}_X(ui))$, and $b \in \text{Ind}(\mathbf{T}_X(uj))$ with $[a]_{\mathcal{P}} = [b]_{\mathcal{P}}$. Then Claim (2) yields $g_X(ui) = a^{\mathcal{I}} = b^{\mathcal{I}} = g_X(uj)$, and thus $\mathbf{T}_X(ui)(\rho) = s^{\mathcal{I}}(g_X(u), a^{\mathcal{I}}) = \mathbf{T}_X(uj)(\rho)$.

For Condition (iv) consider $u \in K^*$, $a \in \text{Ind}(\mathbf{T}_X(u))$, $s \in \text{Rol}^-(\mathcal{O})$, $i \in \varphi_s(\mathcal{O})$, and $b \in \text{Ind}(\mathbf{T}_X(ui))$. By Claim (2), we get $g_X(u) = a^{\mathcal{I}}$, $g_X(ui) = b^{\mathcal{I}}$, and $\mathbf{T}_X(ui)(\rho) = s^{\mathcal{I}}(g_X(u), g_X(ui)) = s^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) = s_{\mathcal{P}}([a]_{\mathcal{P}}, [b]_{\mathcal{P}})$. \square

From this lemma it follows that consistency in $L\text{-}\mathcal{ALCCOI}$ can be reduced to deciding the existence of suitable family of Hintikka trees. By building looping automata whose runs correspond exactly to those Hintikka trees, we further reduce it to the emptiness problem for this class of automata.

Definition 18 (Hintikka automaton). *Let $(\mathcal{P}, \mathcal{H}_{\mathcal{P}}, \mathcal{R}_{\mathcal{P}})$ be a pre-completion for \mathcal{O} and $X \in \mathcal{P}$. The Hintikka automaton for \mathcal{O} and H_X is the looping automaton $\mathbf{A}_{\mathcal{O}, H_X} = (Q_{\mathcal{O}}, I_{\mathcal{O}, H_X}, \Delta_{\mathcal{O}})$, where*

- $Q_{\mathcal{O}}$ is the set of all Hintikka functions for \mathcal{O} that are compatible with \mathcal{T} and the pre-completion,
- $I_{\mathcal{O}, H_X} := \{H_X\}$, and
- $\Delta_{\mathcal{O}}$ is the set of all elements of $Q_{\mathcal{O}}^{k+1}$ that satisfy the Hintikka condition.

The runs of $\mathbf{A}_{\mathcal{O}, H_X}$ are exactly the Hintikka trees for \mathcal{O} starting with H_X . Thus, \mathcal{O} is consistent iff there is a pre-completion $(\mathcal{P}, \mathcal{H}_{\mathcal{P}}, \mathcal{R}_{\mathcal{P}})$ for \mathcal{O} such that $\mathbf{A}_{\mathcal{O}, H_X}$ is not empty for all $X \in \mathcal{P}$.

Note that the number of partitions of $\text{Ind}(\mathcal{O})$ is bounded by $2^{|\text{Ind}(\mathcal{O})|^2}$, the number of Hintikka functions for \mathcal{O} is bounded by $(|L|+1)^{|\text{sub}(\mathcal{O})|+1}$, and the number of fuzzy binary relations on a partition of $\text{Ind}(\mathcal{O})$ is bounded by $|L|^{|\text{Ind}(\mathcal{O})|^2}$. Thus, the number of pre-completions for \mathcal{O} is bounded exponentially in the size of \mathcal{O} and polynomially in the size of L . However, each pre-completion is

only of size polynomial in the size of the input. We can thus enumerate all pre-completions in exponential time and for each of them check emptiness of polynomially many looping automata. Since the size of these automata is exponential in the size of \mathcal{O} and emptiness of looping automata is decidable in polynomial time in the size of the automaton [37], the overall runtime of this algorithm is bounded exponentially in the size of \mathcal{O} .

This gives a tight upper bound for the complexity of consistency in $L\text{-}\mathcal{ALCCOI}$ since this problem is already EXPTIME -hard for classical \mathcal{ALCCOI} , even for empty TBoxes [30, 35].

Theorem 19. *In all fuzzy DLs between $L\text{-}\mathcal{ALC}$ and $L\text{-}\mathcal{ALCCOI}$, deciding consistency w.r.t. general TBoxes is EXPTIME -complete.*

When restricting to acyclic TBoxes, reasoning in classical \mathcal{ALCO} is PSPACE -complete [3, 31]. We show in the following section that this remains true under finite lattice semantics. A similar approach was used in [17] to prove the same for $L\text{-}\mathcal{ALCT}$, even in the presence of a role hierarchy or (crisp) transitive roles.

5 Consistency w.r.t. Acyclic TBoxes

Consider now an ontology $\mathcal{O} = (\mathcal{A}, \mathcal{T})$ from $L\text{-}\mathcal{ALCO}$, i.e. that does not use inverse roles, where \mathcal{T} is an acyclic TBox. Notice that we can guess a partition \mathcal{P} of $\text{Ind}(\mathcal{O})$ and families $(H_X)_{X \in \mathcal{P}}$ and $(r_{\mathcal{P}})_{r \in \text{Rol}(\mathcal{O})}$ and verify the conditions of Definition 14 in (non-deterministic) polynomial space. Thus, if we can show that emptiness of the polynomially many Hintikka automata $A_{\mathcal{O}, H_X}$ can be decided in polynomial space, then we would obtain a PSPACE upper bound for deciding consistency in this logic. The idea is to modify the construction of these automata into a PSPACE on-the-fly construction. It is easy to see that the automata $A_{\mathcal{O}, H_X}$ already satisfy all but one of the conditions from Definition 5:

- (i) the arity k of the automata is given by the number of existential and value restrictions in $\text{sub}(\mathcal{O})$;
- (ii) every Hintikka function (i.e. every state of the automaton) has size bounded by $|L|(|\text{sub}(\mathcal{O})| + 1)$ since it consists of $|\text{sub}(\mathcal{O})| + 1$ lattice values;
- (iii) building a state or a transition of the automaton requires only to guess $|\text{sub}(\mathcal{O})| + 1$ or $k(|\text{sub}(\mathcal{O})| + 1)$ lattice values, respectively, and then verifying that this is indeed a valid state or transition of the automaton, which can be done in time polynomial in the size of L and \mathcal{O} .

However, it is possible to build runs of $A_{\mathcal{O}, H_X}$ where blocking occurs only after exponentially many transitions, violating the first condition of PSPACE on-the-fly constructions. We will use a faithful family of functions to obtain reduced automata that guarantee blocking after at most polynomially many transitions, thus obtaining the PSPACE upper bound.

For our construction to work, we need to make a small change to the definition of compatibility of a Hintikka function H with the pre-completion $(\mathcal{P}, \mathcal{H}_{\mathcal{P}}, \mathcal{R}_{\mathcal{P}})$

that we have guessed: for every $a \in \text{Ind}(H)$, we only require the following weaker condition: For all $C \in \text{sub}(\mathcal{O})$ for which $H(C)$ is defined, $H_{[a]_{\mathcal{P}}}(C)$ is also defined and we have $H(C) = H_{[a]_{\mathcal{P}}}(C)$.

It can be seen that this condition would yield an incorrect construction in the presence of inverse roles. However, if inverse roles are disallowed, then all results obtained so far remain true under this modification. More precisely, the only changes in the proof of Lemma 17 are in two places that refer to the compatibility with the pre-completion (without using inverse roles). These are part of the proof of Claim (1) for existential and value restrictions. In both places, it is enough to be able to infer from the fact that a Hintikka function H is compatible with the pre-completion, $a \in \text{Ind}(H)$, and $C \in \text{supp}(H)$ that we have $C \in \text{supp}(H_{[a]_{\mathcal{P}}})$ and $H_{[a]_{\mathcal{P}}}(C) = H(C)$. This is precisely the new condition that we are now considering.

We now describe a faithful family of functions for $\mathbf{A}_{\mathcal{O}, H_X}$ that allows us to obtain a PSPACE-on-the-fly construction. The idea is that it suffices to consider transitions that reduce the maximal role depth (w.r.t. \mathcal{T}) in the support of the states. The *role depth w.r.t. \mathcal{T}* ($\text{rd}_{\mathcal{T}}$) of concepts is recursively defined as follows:

- $\text{rd}_{\mathcal{T}}(A) := \text{rd}_{\mathcal{T}}(\top) := \text{rd}_{\mathcal{T}}(\{a\}) := 0$ for every primitive concept name A and every $a \in \mathbf{N}_I$;
- $\text{rd}_{\mathcal{T}}(A) := \text{rd}_{\mathcal{T}}(C)$ for every definition $\langle A \doteq C \geq \ell \rangle \in \mathcal{T}$;
- $\text{rd}_{\mathcal{T}}(\neg C) := \text{rd}_{\mathcal{T}}(C)$;
- $\text{rd}_{\mathcal{T}}(C \sqcap D) := \text{rd}_{\mathcal{T}}(C \rightarrow D) := \max\{\text{rd}_{\mathcal{T}}(C), \text{rd}_{\mathcal{T}}(D)\}$; and
- $\text{rd}_{\mathcal{T}}(\exists r.C) := \text{rd}_{\mathcal{T}}(\forall r.C) := \text{rd}_{\mathcal{T}}(C) + 1$.

This is well-defined since \mathcal{T} is an acyclic TBox.

Given a Hintikka function H for \mathcal{O} , we define $\text{rd}_{\mathcal{T}}(H)$ as the maximal role depth $\text{rd}_{\mathcal{T}}(C)$ of a concept $C \in \text{supp}(H)$. For $n \geq 0$, we further denote by $\text{sub}^{\leq n}(\mathcal{O})$ the set of all concepts $C \in \text{sub}(\mathcal{O})$ with $\text{rd}_{\mathcal{T}}(C) \leq n$.

Definition 20 (functions f_H). *Let H and H' be two states of $\mathbf{A}_{\mathcal{O}, H_X}$ and $n := \text{rd}_{\mathcal{T}}(H)$. We define the function $f_H(H')$ as follows for every $C \in \text{sub}(\mathcal{O})$:*

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

$$f_H(H')(\rho) := \begin{cases} \mathbf{0} & \text{if } \text{supp}(H) = \emptyset, \\ H'(\rho) & \text{otherwise.} \end{cases}$$

Since \mathcal{T} is acyclic, the function $f_H(H')$ defined above is still a Hintikka function for \mathcal{O} compatible with all axioms of \mathcal{T} . It also remains compatible with the pre-completion (according to the modified definition of this notion) since it only discards some values from H' . Thus, it is again an element of $Q_{\mathcal{O}}$.

Lemma 21. *In $L\text{-}\mathcal{ALCO}$, the family of functions f_H is faithful w.r.t. $\mathbf{A}_{\mathcal{O}, H_X}$.*

Proof. Consider $H, H_0, H_1, \dots, H_k \in Q_{\mathcal{O}}$ and let $n := \text{rd}_{\mathcal{T}}(H)$ and $H'_i := f_H(H_i)$ for all i , $0 \leq i \leq k$. We first verify that if (H, H_1, \dots, H_k) satisfies the Hintikka condition, then (H, H'_1, \dots, H'_k) also satisfies it.

If $\exists s.C \in \text{supp}(H)$, then $C \in \text{supp}(H_i)$ and $H(\exists s.C) = H_i(\rho) \otimes H_i(C)$, where $i := \varphi(\exists s.C)$. Since we have $\text{rd}_{\mathcal{T}}(C) < \text{rd}_{\mathcal{T}}(\exists s.C) \leq \text{rd}_{\mathcal{T}}(H) = n$, this implies that $C \in \text{supp}(H'_i)$ and $H'_i(C) = H_i(C)$. Moreover, we know that $\text{supp}(H) \neq \emptyset$, and thus $H'_i(\rho) = H_i(\rho)$. This shows that the required equality $H(\exists s.C) = H_i(\rho) \otimes H_i(C) = H'_i(\rho) \otimes H'_i(C)$ remains satisfied.

Let now $\exists s.C \in \text{supp}(H)$ and $i \in \varphi_s(\mathcal{O})$. By the same arguments as above, $H'_i(C)$ and $H'_i(\rho)$ are defined and equal to $H_i(C)$ and $H_i(\rho)$, respectively. Thus, the required inequality is still satisfied after applying f_H . Since in $L\text{-ALCCO}$ there are no inverse roles, the rest of Condition (i) of Definition 15 is trivially satisfied. Condition (ii) follows by similar arguments.

For Condition (iii), consider any role name $r \in \text{Rol}(\mathcal{O})$ and $i, j \in \varphi_s(\mathcal{O})$. If there are $a \in \text{Ind}(H'_i)$ and $b \in \text{Ind}(H'_j)$ with $[a]_{\mathcal{P}} = [b]_{\mathcal{P}}$, then this must already have been the case for H_i and H_j . Since then $\text{supp}(H)$ cannot be empty, we still have $H'_i(\rho) = H_i(\rho) = H_j(\rho) = H'_j(\rho)$. A similar argument shows that Condition (iv) remains satisfied.

For the second condition of Definition 3, we assume that (H_0, H_1, \dots, H_k) satisfies the Hintikka condition and verify it for $(H'_0, H'_1, \dots, H'_k)$.

Consider any $\exists s.C \in \text{supp}(H'_0)$ and $i := \varphi(\exists s.C)$. By the definition of f_H , we get $H_0(\exists s.C) = H'_0(\exists s.C)$ and $\text{rd}_{\mathcal{T}}(C) < \text{rd}_{\mathcal{T}}(\exists s.C) < \text{rd}_{\mathcal{T}}(H)$. Thus, $H_i(C)$ is defined and equal to $H'_i(C)$. Moreover, $\text{supp}(H) \neq \emptyset$, which implies that $H'_i(\rho) = H_i(\rho)$. Thus, $H'_0(\exists s.C) = H_0(\exists s.C) = H_i(\rho) \otimes H_i(C) = H'_i(\rho) \otimes H'_i(C)$.

Again, the remaining part of Condition (i) can be shown by similar arguments, replacing $\varphi(\exists s.C)$ by an element of $\varphi_s(\mathcal{O})$ and the equality condition by an inequality. The remaining conditions are similarly easy to verify. \square

By Lemma 4, $\mathbf{A}_{\mathcal{O}, H_X}$ is empty iff the induced subautomaton $\mathbf{A}_{\mathcal{O}, H_X}^S$ is empty. It remains to show that the latter problem can be decided in PSPACE.

Theorem 22. *The construction of $\mathbf{A}_{\mathcal{O}, H_X}^S$ from L , \mathcal{O} , and H_X is a PSPACE on-the-fly construction.*

Proof. We show that the automata $\mathbf{A}_{\mathcal{O}, H_X}^S$ are m -blocking for

$$m := \max\{\text{rd}_{\mathcal{T}}(C) \mid C \in \text{sub}(\mathcal{O})\} + 3.$$

The other conditions of Definition 5 have already been shown above.

By the definition of $\mathbf{A}_{\mathcal{O}, H_X}^S$, every transition decreases the maximal role depth of the support of the state. Hence, after at most $\max\{\text{rd}_{\mathcal{T}}(C) \mid C \in \text{sub}(\mathcal{O})\} + 1$ transitions, we must reach a state H that is undefined for all $C \in \text{sub}(\mathcal{O})$, and hence $\text{supp}(H) = \emptyset$. From the next transition on, all states additionally assign $\mathbf{0}$ to ρ . Hence, after at most m transitions, we find two states that are equal. Since m is bounded by a polynomial in the size of \mathcal{O} , the automata $\mathbf{A}_{\mathcal{O}, H_X}^S$ satisfy Definition 5. \square

Theorem 6 yields the desired PSPACE upper bound for consistency in $L\text{-ALCCO}$. PSPACE-hardness follows from PSPACE-hardness of consistency w.r.t. the empty TBox in classical ALC [31].

Theorem 23. *In $L\text{-}\mathcal{ALCCO}$, the problem of deciding consistency w.r.t. acyclic TBoxes is PSPACE-complete.*

Using a different faithful family of functions, it was shown in [11] that consistency of $L\text{-}\mathcal{ALCI}$ ontologies with acyclic TBoxes is also PSPACE-complete. As for \mathcal{ALCCO} , this matches the complexity of reasoning in classical DLs.

Notice that the notions of Hintikka functions and Hintikka trees are independent of the operators used. One could use the residual negation $\ominus \ell := \ell \Rightarrow \mathbf{0}$ to interpret the constructor \neg , or the Kleene-Dienes implication $\ell_1 \Rightarrow \ell_2 := \sim \ell_1 \vee \ell_2$ instead of the residuum. The only restrictions are that the semantics must be truth functional, i.e. the value of a formula must depend only on the values of its direct subformulae, and the underlying operators must be computable in polynomial time from the lattice values. We could also use a slightly different semantics for concept definitions in which \otimes is replaced by the simple meet t-norm \wedge .

The algorithm can be modified for reasoning w.r.t. n -witnessed models for $n > 1$. One needs only extend the arity of the Hintikka trees to account for n witnesses for each quantified concept in $\text{sub}(\mathcal{O})$; the arity of $A_{\mathcal{O}, H_X}$ then grows polynomially in n . This does not affect the obtained complexity upper bounds, and hence Theorems 19 and 23 still hold.

6 Conclusions

We have shown that reasoning in $L\text{-}\mathcal{ALCCOI}$ is not harder than in the underlying crisp DL \mathcal{ALCCOI} , if L is a finite De Morgan lattice. More precisely, all the standard reasoning problems in this logic are EXPTIME-complete, even if the TBox is assumed to be empty. If we disallow either nominals or inverse roles, obtaining the logics $L\text{-}\mathcal{ALCI}$ and $L\text{-}\mathcal{ALCCO}$, respectively, then reasoning w.r.t. acyclic TBoxes is PSPACE-complete.

These complexity bounds extend previously known results for lattice-based fuzzy DLs [11, 13, 14, 18] and complements the work in [17], in which $L\text{-}\mathcal{ALCI}$ is extended to allow transitivity and role inclusion axioms. Thus, tight complexity bounds are also obtained for logics up to $L\text{-}\mathcal{SHI}$, under some restrictions in the interpretation of roles. Our methods demonstrate, once again, that automata can show PSPACE results for (fuzzy) DLs [2].

It is reasonable to expect that the construction from [17] for $L\text{-}\mathcal{SHI}$ can be combined with the ideas from this paper for handling nominals, to obtain an automata-based algorithm for reasoning in $L\text{-}\mathcal{SHOI}$. A missing step is to further generalize these methods, or develop new ones, to prove tight complexity bounds for fuzzy variants of the current standard ontology languages, like $\mathcal{SROIQ}(\mathcal{D})$. We also need to understand the effect of removing the restrictions on roles from [17] to the complexity of reasoning.

Although their run-time behavior is optimal w.r.t. the complexity of the problem, automata-based methods are typically not used in practice since their *best-case* behavior is as bad as in the worst-case. It would thus be desirable to produce reasoning algorithms that preserve the properties of the algorithms used

by current classical reasoners [6,28,32]. First steps in this direction have been made in [12,15], where tableau-based algorithms with better run-time behavior are proposed. Those algorithms, however, require a high-level of non-determinism and are thus inappropriate for efficient implementation. Ideas for improvements will be studied.

References

1. Baader, F., Borgwardt, S., Peñaloza, R.: On the decidability status of fuzzy \mathcal{ALC} with general concept inclusions. *J. Philos. Logic* (2014). doi:[10.1007/s10992-014-9329-3](https://doi.org/10.1007/s10992-014-9329-3)
2. Baader, F., Hladik, J., Peñaloza, R.: Automata can show PSPACE results for description logics. *Inf. Comput.* **206**(9–10), 1045–1056 (2008)
3. 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)
4. Baader, F., Peñaloza, R.: Are fuzzy description logics with general concept inclusion axioms decidable? In: *Proceedings of the 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'11)*, pp. 1735–1742. IEEE Computer Society Press (2011)
5. Baader, F., Peñaloza, R.: On the undecidability of fuzzy description logics with GCIs and product T-norm. In: Tinelli, C., Sofronie-Stokkermans, V. (eds.) *FroCoS 2011*. LNCS, vol. 6989, pp. 55–70. Springer, Heidelberg (2011)
6. Baader, F., Sattler, U.: An overview of tableau algorithms for description logics. *Stud. Logica* **69**(1), 5–40 (2001)
7. Bobillo, F., Bou, F., Straccia, U.: On the failure of the finite model property in some fuzzy description logics. *Fuzzy Sets Syst.* **172**(1), 1–12 (2011)
8. Bobillo, F., Straccia, U.: A fuzzy description logic with product t-norm. In: *Proceedings of the 2007 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'07)*, pp. 1–6. IEEE Computer Society Press (2007)
9. Bobillo, F., Straccia, U.: Fuzzy description logics with general t-norms and datatypes. *Fuzzy Sets Syst.* **160**(23), 3382–3402 (2009)
10. Bobillo, F., Straccia, U.: Reasoning with the finitely many-valued Lukasiewicz fuzzy description logic \mathcal{SROIQ} . *Inf. Sci.* **181**(4), 758–778 (2011)
11. Borgwardt, S., Peñaloza, R.: Finite lattices do not make reasoning in \mathcal{ALCI} harder. In: Bobillo, F., et al. (eds.) *Proceedings of the 7th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW'11)*. CEUR Workshop Proceedings, vol. 778, pp. 51–62 (2011)
12. Borgwardt, S., Peñaloza, R.: Consistency reasoning in lattice-based fuzzy description logics. *Int. J. Approximate Reasoning* **55**(9), 1917–1938 (2014)
13. Borgwardt, S., Peñaloza, R.: Description logics over lattices with multi-valued ontologies. In: Walsh, T. (ed.) *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, pp. 768–773. AAAI Press (2011)
14. Borgwardt, S., Peñaloza, R.: Fuzzy ontologies over lattices with t-norms. In: Rosati, R., Rudolph, S., Zakharyashev, M. (eds.) *Proceedings of the 2011 International Workshop on Description Logics (DL'11)*, pp. 70–80. CEUR Workshop Proceedings (2011)
15. Borgwardt, S., Peñaloza, R.: A tableau algorithm for fuzzy description logics over residuated De Morgan lattices. In: Krötzsch, M., Straccia, U. (eds.) *RR 2012*. LNCS, vol. 7497, pp. 9–24. Springer, Heidelberg (2012)

16. Borgwardt, S., Peñaloza, R.: Undecidability of fuzzy description logics. In: Brewka, G., Eiter, T., McClraith, S.A. (eds.) Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR'12), pp. 232–242. AAAI Press (2012)
17. Borgwardt, S., Peñaloza, R.: The complexity of lattice-based fuzzy description logics. *J. Data Seman.* **2**(1), 1–19 (2013)
18. Bou, F., Cerami, M., Esteva, F.: Finite-valued Lukasiewicz modal logic is PSPACE-complete. In: Walsh, T. (ed.) Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11), pp. 774–779. AAAI Press (2011)
19. Cerami, M., García-Cerdaña, A., Esteva, F.: On finitely-valued fuzzy description logics. *Int. J. Approximate Reasoning* **55**(9), 1890–1916 (2014)
20. Cerami, M., Straccia, U.: On the (un)decidability of fuzzy description logics under Lukasiewicz t-norm. *Inf. Sci.* **227**, 1–21 (2013)
21. De Cooman, G., Kerre, E.E.: Order norms on bounded partially ordered sets. *J. Fuzzy Math.* **2**, 281–310 (1993)
22. 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, Amsterdam (2007)
23. Hájek, P.: *Metamathematics of Fuzzy Logic (Trends in Logic)*. Springer, London (2001)
24. Hájek, P.: Making fuzzy description logic more general. *Fuzzy Sets Syst.* **154**(1), 1–15 (2005)
25. Horrocks, I., Sattler, U.: A description logic with transitive and inverse roles and role hierarchies. *J. Logic Comput.* **9**(3), 385–410 (1999)
26. Klement, E.P., Mesiar, R., Pap, E.: *Triangular Norms*. Trends in Logic, Studia Logica Library. Springer, New York (2000)
27. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in description logics for the semantic web. *J. Web Seman.* **6**(4), 291–308 (2008)
28. Motik, B., Shearer, R., Horrocks, I.: Hypertableau reasoning for description logics. *J. Artif. Intell. Res.* **36**, 165–228 (2009)
29. Schaerf, A.: Reasoning with individuals in concept languages. *Data Knowl. Eng.* **13**(2), 141–176 (1994)
30. Schild, K.: A correspondence theory for terminological logics: preliminary report. In: Mylopoulos, J., Reiter, R. (eds.) Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI'91), pp. 466–471. Morgan Kaufmann (1991)
31. Schmidt-Schauß, M., Smolka, G.: Attributive concept descriptions with complements. *Artif. Intell.* **48**(1), 1–26 (1991)
32. Steigmiller, A., Liebig, T., Glimm, B.: Extended caching, backjumping and merging for expressive description logics. In: Gramlich, B., Miller, D., Sattler, U. (eds.) IJCAR 2012. LNCS, vol. 7364, pp. 514–529. Springer, Heidelberg (2012)
33. Straccia, U.: Description logics over lattices. *Int. J. Uncertainty Fuzziness Knowl.-Based Syst.* **14**(1), 1–16 (2006)
34. Straccia, U.: *Foundations of Fuzzy Logic and Semantic Web Languages*. Studies in Informatics. CRC Press, Hoboken (2013)
35. Tobies, S.: The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *J. Artif. Intell. Res.* **12**, 199–217 (2000)
36. Tobies, S.: Complexity results and practical algorithms for logics in knowledge representation. Ph.D. thesis, RWTH Aachen, Germany (2001)
37. Vardi, M.Y., Wolper, P.: Automata-theoretic techniques for modal logics of programs. *J. Comput. Syst. Sci.* **32**(2), 183–221 (1986)