The Precise Complexity of Reasoning in $\mathcal{ALC}$ with ω-Admissible Concrete Domains

Stefan Borgwardt[1], Filippo De Bortoli[1,2] and Patrick Koopmann[3]

[1]TU Dresden, Institute of Theoretical Computer Science
Dresden, Germany
[2]Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI)
Dresden/Leipzig, Germany
[3]Vrije Universiteit Amsterdam, Department of Computer Science
Amsterdam, The Netherlands

Abstract

Concrete domains have been introduced in the context of Description Logics to allow references to qualitative and quantitative values. In particular, the class of ω-admissible concrete domains, which includes Allen's interval algebra, the region connection calculus (RCC8), and the rational numbers with ordering and equality, has been shown to yield extensions of $\mathcal{ALC}$ for which concept satisfiability w.r.t. a general TBox is decidable. In this paper, we present an algorithm based on type elimination and use it to show that deciding the consistency of an $\mathcal{ALC}(\mathfrak{D})$ ontology is ExpTime-complete if the concrete domain $\mathfrak{D}$ is ω-admissible and its constraint satisfaction problem is decidable in exponential time. While this allows us to reason with concept and role assertions, we also investigate *feature assertions* $f(a, c)$ that can specify a constant c as the value of a feature f for an individual a. We show that, under conditions satisfied by all known ω-admissible domains, we can add feature assertions without affecting the complexity.

Keywords

Description Logics, Concrete Domains, Reasoning, Complexity, Type Elimination

1. Introduction

Reasoning about numerical attributes of objects is a core requirement of many applications. For this reason, *Description Logics (DLs)* that integrate reasoning over an abstract domain of knowledge with references to values drawn from a *concrete domain* have already been investigated for over 30 years [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. In this setting, DLs are extended by *concrete features* that are interpreted as partial functions mapping abstract domain elements to concrete values, such as the feature diastolic that describes the diastolic blood pressure of a person. Using *concrete domain restrictions* we can then describe constraints over these features and state, e.g. that all patients have a diastolic blood pressure that is lower than their systolic blood pressure, by adding the concept inclusion $\mathsf{Patient} \sqsubseteq \exists \mathsf{diastolic}, \mathsf{systolic}.{<}$ to the ontology. More interestingly, we can use *feature paths* to compare the feature values of different individuals: the inclusion $\top \sqsubseteq \forall \mathsf{hasChild}\ \mathsf{age}, \mathsf{age}.{<}$ states that children are always younger than their parents.

Unfortunately, dealing with feature paths in DLs is very challenging and often leads to undecidability [2, 11, 9], which is why restrictive conditions on the concrete domains were introduced to regain decidability. An approach inspired from research on constraint satisfaction is based on ω-*admissibility* [5], which requires certain compositionality properties for finite and countable sets of constraints. This enables the composition of full models from local solutions of sets of concrete domain constraints. Being quite restrictive, at first only two examples of ω-admissible concrete domains were

DL 2024: 37th International Workshop on Description Logics, June 18–21, 2024, Bergen, Norway
stefan.borgwardt@tu-dresden.de (S. Borgwardt); filippo.de_bortoli@tu-dresden.de (F. De Bortoli); p.k.koopmann@vu.nl (P. Koopmann)
https://lat.inf.tu-dresden.de/~stefborg (S. Borgwardt); https://lat.inf.tu-dresden.de/~debortoli (F. De Bortoli); https://pkoopmann.github.io (P. Koopmann)
0000-0003-0924-8478 (S. Borgwardt); 0000-0002-8623-6465 (F. De Bortoli); 0000-0001-5999-2583 (P. Koopmann)

known, namely Allen's interval algebra and the region connection calculus (RCC8). Recently, the conditions of ω-admissibility were investigated in more detail, and it was shown that ω-admissible concrete domains can be obtained from *finitely bounded homogeneous structures* [9]. This class includes the rational numbers $\mathfrak{Q} := (\mathbb{Q}, <, =, >)$ with ordering and equality, and is closed under union and certain types of product [9].

Despite these extensive investigations, the precise complexity of reasoning using description logics with concrete domains has been established only for a few special cases so far [2, 7, 8, 10]. These results have been obtained by automata-based techniques, and yield decidability or tight complexity bounds for cases where the concrete domain is not ω-admissible, such as the ordered integers $(\mathbb{Z}, <, =, >)$ [8] or the set of strings over a finite alphabet with a prefix relation [10]. In contrast, the goal of this paper is to derive a tight complexity bound for ontology consistency in the extension of $\mathcal{ALC}$ with any ω-admissible concrete domain. Using an algorithm based on type elimination, we establish that this decision problem is ExpTime-complete under the assumption that the concrete domain $\mathfrak{D}$ is ω-admissible and that its *constraint satisfaction problem* (CSP) is decidable in exponential time. The latter condition holds for all finitely bounded homogeneous structures (including the three examples from above), since their CSP is guaranteed to be in NP and in some cases, e.g. for $\mathfrak{Q}$, is decidable in polynomial time.

Additionally, we consider *predicate* and *feature assertions*. These allow us to constrain values of features associated to named individuals, e.g. to write $<(\mathsf{age}(\mathsf{mary}), \mathsf{age}(\mathsf{bob}))$ to state that Mary is younger than Bob, or to specify a constant value of a feature for a named individual, e.g. using $\mathsf{systolic}(\mathsf{mary}, 122)$ to state that Mary's systolic blood pressure equals 122. We prove that consistency in the presence of predicate assertions can always be reduced in polynomial time to consistency without predicate assertions. For feature assertions, we show that this holds if we consider *homogeneous* ω-admissible concrete domains, which again is the case for the three concrete domains from above. Finally, we show that supporting feature assertions is equivalent to supporting singleton predicates of the form $=_c$ for a constant c in the concrete domain.

Omitted proofs of the results presented in the text can be found in [12].

2. Preliminaries

Concrete Domains. As usual for DLs, we adopt the term *concrete domain* to refer to a relational structure $\mathfrak{D} = (D, P_1^D, P_2^D, \dots)$ over a non-empty, countable relational signature $\{P_1, P_2, \dots\}$, where D is a non-empty set, and each predicate P has an associated arity $k \in \mathbb{N}$ and is interpreted by a relation $P^D \subseteq D^k$. An example is the structure $\mathfrak{Q} := (\mathbb{Q}, <, =, >)$ over the rational numbers $\mathbb{Q}$ with standard binary order and equality relations. Given a countably infinite set V of variables, a *constraint system* over V is a set $\mathfrak{C}$ of *constraints* $P(v_1, \dots, v_k)$, where $v_1, \dots, v_k \in V$ and P is a k-ary predicate of $\mathfrak{D}$. We denote by $V(\mathfrak{C})$ the set of variables that occur in $\mathfrak{C}$. The constraint system $\mathfrak{C}$ is *satisfiable* if there is a *homomorphism* $h\colon V(\mathfrak{C}) \to D$ that satisfies every constraint in $\mathfrak{C}$, i.e. $P(v_1, \dots, v_k) \in \mathfrak{C}$ implies $(h(v_1), \dots, h(v_k)) \in P^D$. We then call h a *solution* of $\mathfrak{C}$. The *constraint satisfaction problem* for $\mathfrak{D}$, denoted $\mathrm{CSP}(\mathfrak{D})$, is the decision problem asking whether a finite constraint system $\mathfrak{C}$ over $\mathfrak{D}$ is satisfiable. The problem $\mathrm{CSP}(\mathfrak{Q})$ is in P, since satisfiability can be reduced to $<$-cycle detection; for example, the 3-clique $\{x_1 < x_2,\ x_2 < x_3,\ x_3 < x_1\}$ (using infix notation for $<$) is unsatisfiable over $\mathfrak{Q}$.

To ensure that reasoning in the extension of $\mathcal{ALC}$ with concrete domain restrictions is decidable, we impose further properties on $\mathfrak{D}$ regarding its relations and the compositionality of its CSP for finite and countable constraint systems. We say that $\mathfrak{D}$ is a *patchwork* if it satisfies the following conditions:[1]

JEPD for all $k \geq 1$, either $\mathfrak{D}$ has no k-ary relation, or D^k is partitioned by all k-ary relations;

JD there is a quantifier-free, equality-free first-order formula over the signature of $\mathfrak{D}$ that defines the equality relation $=$ between two elements of $\mathfrak{D}$;

[1] [5] originally used only JEPD (jointly exhaustive, pairwise disjoint) and AP (amalgamation property). JD (jointly diagonal) was later added by [9].

AP if $\mathfrak{B}$, $\mathfrak{C}$ are constraint systems and $P(v_1, \ldots, v_k) \in \mathfrak{B}$ iff $P(v_1, \ldots, v_k) \in \mathfrak{C}$ holds for all $v_1, \ldots, v_k \in V(\mathfrak{B}) \cap V(\mathfrak{C})$ and all k-ary predicates P over $\mathfrak{D}$, then $\mathfrak{B}$ and $\mathfrak{C}$ are satisfiable iff $\mathfrak{B} \cup \mathfrak{C}$ is satisfiable.

If $\mathfrak{D}$ is a patchwork, we call a constraint system $\mathfrak{C}$ *complete* if, for all $k \in \mathbb{N}$, either $\mathfrak{D}$ has no k-ary predicates, or for all $v_1, \ldots, v_k \in V(\mathfrak{C})$ there is exactly one k-ary predicate P over $\mathfrak{D}$ such that $P(v_1, \ldots, v_k) \in \mathfrak{C}$. The concrete domain $\mathfrak{D}$ is *homomorphism ω-compact* if a countable constraint system $\mathfrak{C}$ over $\mathfrak{D}$ is satisfiable whenever every finite constraint system $\mathfrak{C}' \subseteq \mathfrak{C}$ is satisfiable. We now introduce our definition of ExpTime-ω-admissible concrete domains, which differs from the definitions of ω-admissibility [5, 9] in that we require $\mathrm{CSP}(\mathfrak{D})$ to be decidable in exponential time instead of simply decidable.

Definition 1. *A concrete domain* $\mathfrak{D}$ *is* ExpTime-ω-admissible *if*

- $\mathfrak{D}$ *has a finite signature,*
- $\mathfrak{D}$ *is a patchwork,*
- $\mathfrak{D}$ *is homomorphism ω-compact, and*
- $\mathrm{CSP}(\mathfrak{D})$ *is decidable in exponential time.*

Requiring the signature of $\mathfrak{D}$ to be finite is necessary to ensure decidability of $\mathcal{ALC}(\mathfrak{D})$ [9]. In Section 4, we will also consider concrete domains $\mathfrak{D}$ that are *homogeneous*, that is, every isomorphism between finite substructures of $\mathfrak{D}$ can be extended to an isomorphism from $\mathfrak{D}$ to itself. All properties we have defined here are satisfied by the three examples of Allen's interval relations, RCC8, and $\mathfrak{Q}$, as they are *finitely bounded* homogeneous structures, which are ω-admissible and have CSPs whose complexity is at most NP (see [9] for details).

Ontologies with Concrete Domain Constraints. We assume the reader to be familiar with the standard description logic $\mathcal{ALC}$ [13]. To use a concrete domain $\mathfrak{D}$ in $\mathcal{ALC}$ axioms, we introduce a set of *concrete features* $\mathsf{N_F}$, where each $f \in \mathsf{N_F}$ is interpreted as a *partial* function $f^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \to D$ in an interpretation $\mathcal{I}$. A *feature path* is of the form $r_1 \ldots r_n f$, where $r_1, \ldots, r_n \in \mathsf{N_R}$ are role names and $f \in \mathsf{N_F}$. The semantics of such a path is given by a function

$$(r_1 \ldots r_n f)^{\mathcal{I}}(d) \coloneqq \big\{ f^{\mathcal{I}}(e) \mid (d,e) \in r_1^{\mathcal{I}} \circ \cdots \circ r_n^{\mathcal{I}} \text{ and } f^{\mathcal{I}}(e) \text{ is defined} \big\}$$

that assigns to each domain element the set of all f-values of elements e reachable via the role chain $r_1 \ldots r_n$ (which may be empty if there is no such e, or if $f^{\mathcal{I}}(e)$ is always undefined). In a slight abuse of notation, we allow the case where $n = 0$, i.e. f can be seen as both a feature and a feature path, with slightly different, but equivalent semantics (that is, a partial function vs. a set-valued function that may produce either a singleton set or the empty set). $\mathcal{ALC}(\mathfrak{D})$ *concepts* are defined similarly to $\mathcal{ALC}$ concepts, but can additionally use the following concept constructor: A *concrete domain restriction* (or simply *CD-restriction*) is of the form $\exists p_1, \ldots, p_k.P$ or $\forall p_1, \ldots, p_k.P$, where $p_1, \ldots, p_k$ are feature paths and P is a k-ary predicate, with the semantics

$$(\exists p_1, \ldots, p_k.P)^{\mathcal{I}} \coloneqq \big\{ d \in \Delta^{\mathcal{I}} \mid \text{there are } c_i \in p_i^{\mathcal{I}}(d) \text{ for } i = 1, \ldots, k \text{ s.t. } (c_1, \ldots, c_k) \in P^D \big\},$$
$$(\forall p_1, \ldots, p_k.P)^{\mathcal{I}} \coloneqq \big\{ d \in \Delta^{\mathcal{I}} \mid \text{if } c_i \in p_i^{\mathcal{I}}(d) \text{ for } i = 1, \ldots, k, \text{ then } (c_1, \ldots, c_k) \in P^D \big\}.$$

An $\mathcal{ALC}(\mathfrak{D})$ *ontology* $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ consists of an *ABox* $\mathcal{A}$ and a *TBox* $\mathcal{T}$ over $\mathcal{ALC}(\mathfrak{D})$ concepts.

Note that we explicitly consider $\mathcal{ALC}(\mathfrak{D})$ and not $\mathcal{ALCF}(\mathfrak{D})$, i.e. we do not allow functional roles, in contrast to other definitions from the literature [1, 5]. In the literature, feature paths are either restricted to contain only functional roles, or to have a length of at most 2, (compare with [2, 5]). Consequently, we restrict ourselves to feature paths of length ≤ 2, that is, assume that they are of the form f or rf. We are not aware of any work that considers feature paths over non-functional roles of length longer than 2, and leave the investigation of this case for future work. Without loss of generality, we can assume that universal CD-restrictions are not used in concepts, because we can express $\forall p_1, \ldots, p_k.P$

as $\neg\exists p_1,\dots,p_k.P_1 \sqcap \dots \sqcap \neg\exists p_1,\dots,p_k.P_m$, where $P_1,\dots,P_m$ are all k-ary predicates of $\mathfrak{D}$ except for P (the union of these predicates is equivalent to the complement of P due to JEPD, and there are only finitely many of them since the signature is assumed to be finite). We can additionally assume that concepts do not contain value restrictions $\forall r.C$ or disjunctions $C \sqcup D$ since they can be expressed using only negation, conjunction and existential restriction.

3. Consistency in $\mathcal{ALC}(\mathfrak{D})$

Let now $\mathfrak{D}$ be a fixed ExpTime-ω-admissible concrete domain, $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ be an $\mathcal{ALC}(\mathfrak{D})$ ontology, and $\mathcal{M}$ be the set of all subconcepts appearing in $\mathcal{O}$ and their negations. For the type elimination algorithm, we start by defining the central notion of *types*, which is standard.

Definition 2. *A set $t \subseteq \mathcal{M}$ is a* type *w.r.t. $\mathcal{O}$ if it satisfies the following properties:*

- *if $C \sqsubseteq D \in \mathcal{T}$ and $C \in t$, then $D \in t$;*
- *if $\top \in \mathcal{M}$, then $\top \in t$;*
- *if $\neg D \in \mathcal{M}$, then $D \in t$ iff $\neg D \notin t$;*
- *if $D \sqcap D' \in \mathcal{M}$, then $D \sqcap D' \in t$ iff $D \in t$ and $D' \in t$.*

Given a model $\mathcal{I}$ of $\mathcal{O}$ and an individual $d \in \Delta^{\mathcal{I}}$, the type of d *w.r.t. $\mathcal{O}$ is the set*

$$t_{\mathcal{I}}(d) := \{C \in \mathcal{M} \mid d \in C^{\mathcal{I}}\}.$$

Clearly, $t_{\mathcal{I}}(d)$ satisfies the four conditions required to be a type w.r.t. $\mathcal{O}$. We use this connection between individuals and types to define *augmented types* that represent the relationship between an individual, its role successors, and the CD-restrictions that ought to be satisfied. Hereafter, let n_{ex} be the number of existential restrictions $\exists r.C$ in $\mathcal{M}$, and n_{cd} the number of CD-restrictions $\exists p_1,\dots,p_k.P$ in $\mathcal{M}$. The maximal arity of predicates P occurring in $\mathcal{M}$ is denoted by n_{ar}, and we define $n_{\mathcal{O}} := n_{\mathsf{ex}} + n_{\mathsf{cd}} \cdot n_{\mathsf{ar}}$. Intuitively, each non-negated existential restriction in a type needs a successor (and associated type) to be realized, while CD-restrictions may require n_{ar} role successors to fulfill a certain constraint. Therefore, $n_{\mathcal{O}}$ is an upper bound on the number of successors needed to satisfy all the non-negated restrictions occurring in a type t w.r.t. $\mathcal{O}$.

Given a type t_0, we define a constraint system associated with a sequence of types $t_1,\dots,t_{n_{\mathcal{O}}}$ representing the role successors of a domain element with type t_0. This system contains a variable f^i for each feature f of an individual with type t_i in order to express the relevant CD-restrictions. Concrete features that are not represented in this system can remain undefined since their values are irrelevant for satisfying the CD-restrictions.

Definition 3. *A* local system *for a type t_0 w.r.t. a sequence of types $t_1,\dots,t_{n_{\mathcal{O}}}$ is a complete constraint system $\mathfrak{C}$ for which there exists a* successor function $\mathsf{succ}\colon \mathsf{N_R}(\mathcal{O}) \to \mathcal{P}(\{1,\dots,n_{\mathcal{O}}\})$*, such that, for all $\exists p_1,\dots,p_k.P \in \mathcal{M}$, the following condition holds:*

$$\exists p_1,\dots,p_k.P \in t_0 \textit{ iff there is } P(v_1,\dots,v_k) \in \mathfrak{C} \textit{ for some variables } v_1,\dots,v_k \textit{ such that}$$

$$v_i = \begin{cases} f^0 & \textit{if } p_i = f, \textit{ or} \\ f^j & \textit{if } p_i = rf \textit{ and } j \in \mathsf{succ}(r). \end{cases}$$

We use a *sequence* instead of a *set* of types for the role successors, since there can be TBoxes that require the existence of two successors with the same type that only differ in their feature values. For example, for the consistent ontology $\mathcal{O} := \{\top \sqsubseteq \exists rf, rf.<\}$ over $\mathfrak{Q} = (\mathbb{Q}, <, =, >)$, we have $\mathcal{M} = \{\top,\ \neg\top,\ \exists rf, rf.<,\ \neg\exists rf, rf.<\}$, and the only type is $t = \{\top,\ \exists rf, rf.<\}$. Any r-successors witnessing $\exists rf, rf.<$ for an element in a model of $\mathcal{O}$ have the same type t. However, we cannot express the restriction on their f-values by the (unsatisfiable) constraint $<(f^t, f^t)$, but need to consider two copies t_1, t_2 of t to get the (satisfiable) constraint $<(f^1, f^2)$.

To merge the local systems associated to types of adjacent elements in a model, we introduce the following operation. For two local systems $\mathfrak{C}, \mathfrak{C}'$, the *merged system* $\mathfrak{C} \lhd_i \mathfrak{C}'$ is obtained as the union of $\mathfrak{C}$ and $\mathfrak{C}'$ where we identify all features with index i in $\mathfrak{C}$ with those of index 0 in $\mathfrak{C}'$, while keeping the remaining variables separate. Formally, we first replace all variables f^j in $\mathfrak{C}'$ by fresh variables $f^{j'}$ and subsequently replace the variables $f^{0'}$ in $\mathfrak{C}'$ by f^i.

Definition 4. *An* augmented type *for $\mathcal{O}$ is a tuple $\mathfrak{t} := (t_0, \dots, t_{n_\mathcal{O}}, \mathfrak{C}_\mathfrak{t})$ where $t_0, \dots, t_{n_\mathcal{O}}$ are types for $\mathcal{O}$ and $\mathfrak{C}_\mathfrak{t}$ is a local system for t_0 w.r.t. $t_1, \dots, t_{n_\mathcal{O}}$. The* root *of $\mathfrak{t}$ is $\mathsf{root}(\mathfrak{t}) := t_0$. The augmented type $\mathfrak{t}$ is* locally realizable *if $\mathfrak{C}_\mathfrak{t}$ has a solution and if there exists a successor function $\mathsf{succ}_\mathfrak{t}$ for $\mathfrak{C}_\mathfrak{t}$ s.t. for all concepts $\exists r.C \in \mathcal{M}$, it holds that*

$$\exists r.C \in \mathsf{root}(\mathfrak{t}) \textit{ iff there is } i \in \mathsf{succ}_\mathfrak{t}(r) \textit{ such that } C \in t_i.$$

An augmented type $\mathfrak{t}'$ then patches *$\mathfrak{t}$ at $i \in \mathsf{succ}_\mathfrak{t}(r)$ if $\mathsf{root}(\mathfrak{t}') = t_i$ and the system $\mathfrak{C}_\mathfrak{t} \lhd_i \mathfrak{C}_{\mathfrak{t}'}$ has a solution. A set of augmented types $\mathbb{T}$* patches *the locally realizable $\mathfrak{t}$ if, for every role name r and every $i \in \mathsf{succ}_\mathfrak{t}(r)$, there is a $\mathfrak{t}' \in \mathbb{T}$ that patches $\mathfrak{t}$ at i.*

For the ontology $\mathcal{O}$ introduced above, we have $n_\mathcal{O} = 2$ since $\mathcal{M}$ only contains one CD-restriction over a binary predicate. Using infix notation, all augmented types $\mathfrak{t} = (t, t, t, \mathfrak{C}_\mathfrak{t})$ for $\mathcal{O}$ are such that $\mathfrak{C}_\mathfrak{t}$ contains the constraints $f^i = f^i$ for $i = 0, 1, 2$, and either $f^1 < f^2$ or $f^2 < f^1$. There are augmented types $\mathfrak{t}$ that are not locally realizable, for instance if $\mathfrak{C}_\mathfrak{t}$ contains $f^0 = f^1$, $f^0 = f^2$, and $f^1 < f^2$. On the other hand, there is a locally realizable augmented type using the constraints $f^0 < f^1$, $f^1 < f^2$, and $f^0 < f^2$, which can patch itself both at $i \in \{1, 2\}$.

To additionally handle named individuals and concept and role assertions, we introduce a structure $\mathfrak{t}_\mathcal{A}$ that describes all ABox individuals and their connections simultaneously, similar to the common notion of *precompletion*. The associated constraint system $\mathfrak{C}_\mathcal{A}$ now uses variables $f^{a,i}$ indexed with individual names a in addition to numbers i.

Definition 5. *An* ABox type *for $\mathcal{O}$ is a tuple $\mathfrak{t}_\mathcal{A} := \big((\mathfrak{t}_a)_{a \in \mathsf{N_I}(\mathcal{A})}, \mathcal{A}_R, \mathfrak{C}_\mathcal{A}\big)$, where $\mathfrak{t}_a$ are augmented types, $\mathcal{A}_R$ is a set of role assertions over $\mathsf{N_I}(\mathcal{A})$ and $\mathsf{N_R}(\mathcal{O})$, and $\mathfrak{C}_\mathcal{A}$ is a complete and satisfiable constraint system such that, for every $a \in \mathsf{N_I}(\mathcal{A})$,*

- *for every concept assertion $C(a) \in \mathcal{A}$, we have $C \in \mathsf{root}(\mathfrak{t}_a)$;*
- *for every role assertion $r(a, b) \in \mathcal{A}$, we have $r(a, b) \in \mathcal{A}_R$;*
- *for every $\neg\exists r.C \in \mathsf{root}(\mathfrak{t}_a)$ and $r(a, b) \in \mathcal{A}_R$, we have $C \notin \mathsf{root}(\mathfrak{t}_b)$;*
- *for every $P(f_1^{j_1}, \dots, f_k^{j_k}) \in \mathfrak{C}_{\mathfrak{t}_a}$, we have $P(f_1^{a,j_1}, \dots, f_k^{a,j_k}) \in \mathfrak{C}_\mathcal{A}$;*
- *for every $\neg\exists p_1, \dots, p_k.P \in \mathsf{root}(\mathfrak{t}_a)$, there can be no $P(v_1, \dots, v_k) \in \mathfrak{C}_\mathcal{A}$ with*

$$v_i = \begin{cases} f^{a,0} & \textit{if } p_i = f, \\ f^{b,0} & \textit{if } p_i = rf \textit{ and } r(a,b) \in \mathcal{A}_R, \textit{ or} \\ f^{a,j} & \textit{if } p_i = rf \textit{ and } j \in \mathsf{succ}_{\mathfrak{t}_a}(r); \end{cases}$$

Positive occurrences of existential role or CD-restrictions in the ABox type do not need to be handled, as these are satisfied by anonymous successors described in the augmented types $\mathfrak{t}_a$.

3.1. The Type Elimination Algorithm

Algorithm 1 uses the introduced notions to check consistency of $\mathcal{O}$.

Lemma 6 (Soundness). *If Algorithm 1 returns* CONSISTENT*, then $\mathcal{O}$ is consistent.*

Proof. Assume that $\mathbb{T}$ and $\mathfrak{t}_A = ((\mathfrak{t}_a)_{a \in \mathsf{N_I}(\mathcal{A})}, \mathcal{A}_R, \mathfrak{C}_\mathcal{A})$ are obtained after a successful run of the elimination algorithm. We use them to define a forest-shaped interpretation $\mathcal{I}$ that is a model of $\mathcal{O}$. The domain of this model consists of pairs (a, w), where $a \in \mathsf{N_I}$ designates a tree-shaped part of $\mathcal{I}$ whose structure is given by the words w over the alphabet $\Sigma := \mathbb{T} \times \{0, \dots, n_\mathcal{O}\}$. A pair $(\mathfrak{t}, i) \in \Sigma$ describes

Algorithm 1 Elimination algorithm for consistency of $\mathcal{ALC}(\mathfrak{D})$ ontologies

Input: An $\mathcal{ALC}(\mathfrak{D})$ ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$
Output: CONSISTENT if $\mathcal{O}$ is consistent, and INCONSISTENT otherwise

```
M ← all subconcepts occurring in O and their negations
T ← all augmented types for O
while there is t ∈ T that is not locally realizable or not patched by T do
    T ← T \ {t}
if there is an ABox type t_A for O with t_a ∈ T for all a ∈ N_I(A) then
    return CONSISTENT
else
    return INCONSISTENT
```

an augmented type and the position relative to the restriction that this augmented type fulfills w.r.t. its parent in the tree. For a word $w \in \Sigma^+$, we define $\mathsf{end}(w) := \mathsf{t}$ if (t, j) occurs at the last position of w for some $j \in \{0, \dots, n_{\mathcal{O}}\}$.

We start defining the domain of $\mathcal{I}$ by $\Delta^0 := \{(a, w_a) \mid a \in \mathsf{N_I}(\mathcal{A}),\ w_a := (\mathsf{t}_a, 0)\}$. Observe that $w_a \in \Sigma$, since $\mathsf{t}_a \in \mathbb{T}$. Assuming that Δ^m is defined, we define Δ^{m+1} based on Δ^m, and subsequently construct the domain of $\mathcal{I}$ as the union of all sets Δ^m. Given $(a, w) \in \Delta^m$ with $\mathsf{end}(w) = \mathsf{t}$, we observe that t must have a successor function $\mathsf{succ_t}$ s.t. for every $i \in \mathsf{succ_t}(r)$, there is an augmented type $\mathfrak{u}^i \in \mathbb{T}$ patching t at i, as otherwise t would have been eliminated from $\mathbb{T}$. We use these augmented types to define $\Delta_r^{m+1}[a, w] := \{(a, w \cdot (\mathfrak{u}^i, i)) \mid i \in \mathsf{succ_t}(r)\}$ to then obtain

$$\Delta^{m+1} := \Delta^m \cup \bigcup \big\{\Delta_r^{m+1}[a, w] \mid (a, w) \in \Delta^m \text{ and } r \in \mathsf{N_R}\big\}$$

and set $\Delta^{\mathcal{I}} := \bigcup_{m \in \mathbb{N}} \Delta^m$. The interpretation of individual, concept, and role names over $\mathcal{I}$ is given by

$$\begin{aligned}
a^{\mathcal{I}} &:= (a, w_a),\\
A^{\mathcal{I}} &:= \big\{(a, w) \in \Delta^{\mathcal{I}} \mid \mathsf{end}(w) = \mathsf{t} \text{ and } A \in \mathsf{root}(\mathsf{t})\big\},\\
r^{\mathcal{I}} &:= \big\{((a, w_a), (b, w_b)) \mid r(a, b) \in \mathcal{A}_R\big\} \cup\\
&\quad\ \big\{((a, w), (a, w')) \mid (a, w) \in \Delta^m \text{ and } (a, w') \in \Delta_r^{m+1}[a, w] \text{ with } m \in \mathbb{N}\big\}.
\end{aligned}$$

Defining the interpretation of feature names in $\mathcal{I}$ requires more work. Given $(a, w) \in \Delta^{\mathcal{I}}$ with $\mathsf{end}(w) = \mathsf{t}$, let $\mathfrak{C}_{a,w}$ be the constraint system obtained by replacing every variable f^0 in $\mathfrak{C}_\mathsf{t}$ with $f^{a,w}$ and every other variable f^i in $\mathfrak{C}_\mathsf{t}$ with $f^{a,u}$, where $u \in \Sigma^+$ is the unique word of the form $w \cdot (\mathsf{t}', i)$ for which $(a, u) \in \Delta^{\mathcal{I}}$. Correspondingly, let $\mathfrak{C}^0_{\mathcal{A}}$ be the result of replacing all variables $f^{a,0}$ in $\mathfrak{C}_{\mathcal{A}}$ by f^{a,w_a} and $f^{a,i}$ by $f^{a,u}$, where u is the unique word of the form $w_a \cdot (\mathsf{t}', i)$ for which $(a, u) \in \Delta^{\mathcal{I}}$. For $m \in \mathbb{N}$, let $\mathfrak{C}^m$ be the union of $\mathfrak{C}^0_{\mathcal{A}}$ and all constraint systems $\mathfrak{C}_{a,w}$ for which $(a, w) \in \Delta^m$. The proofs of the following claims can be found in [12].

Claim 1. *For every $m \in \mathbb{N}$, the constraint system $\mathfrak{C}^m$ has a solution.*

Using this claim, we show how to define an interpretation of feature names for $\mathcal{I}$. Let $\mathfrak{C}^{\mathcal{I}}$ be the union of all systems $\mathfrak{C}^m$ for $m \in \mathbb{N}$. Every finite system $\mathfrak{B} \subseteq \mathfrak{C}^{\mathcal{I}}$ is also a subsystem of $\mathfrak{C}^m$ for some $m \in \mathbb{N}$. Since $\mathfrak{C}^m$ has a solution, it follows that $\mathfrak{B}$ has a solution. Every finite subsystem of $\mathfrak{C}^{\mathcal{I}}$ has a solution; since $\mathfrak{D}$ has the homomorphism ω-compactness property, we infer that $\mathfrak{C}^{\mathcal{I}}$ has a solution $h^{\mathcal{I}}$. Using this solution, we define for every feature name f the interpretation $f^{\mathcal{I}}(a, w) := h^{\mathcal{I}}(f^{a,w})$ if $f^{a,w}$ occurs in $\mathfrak{C}^{\mathcal{I}}$, and leave it undefined otherwise.

Claim 2. *If $C \in \mathcal{M}$ and $(a, w) \in \Delta^{\mathcal{I}}$ with $\mathsf{end}(w) = \mathsf{t}$, then $C \in \mathsf{root}(\mathsf{t})$ iff $(a, w) \in C^{\mathcal{I}}$.*

By this claim and Definition 2, we know that $\mathcal{I}$ is a model of $\mathcal{T}$. By Definition 5 we obtain that for each $C(a) \in \mathcal{A}$, we have $C \in \mathsf{root}(\mathsf{t}_a)$, and thus $a^{\mathcal{I}} = (a, w_a) = (a, (\mathsf{t}_a, 0)) \in C^{\mathcal{I}}$. Similarly, whenever

$r(a,b) \in \mathcal{A}$, then $r(a,b) \in \mathcal{A}_R$, and thus $(a^\mathcal{I}, b^\mathcal{I}) = ((a, w_a), (b, w_b)) \in r^\mathcal{I}$ by the construction of $\mathcal{I}$. Therefore, we conclude that $\mathcal{I}$ is also a model of $\mathcal{A}$, and thus of $\mathcal{O}$. □

Lemma 7 (Completeness)**.** *If $\mathcal{O}$ is consistent, then Algorithm 1 returns* CONSISTENT*.*

Proof. Let $\mathcal{I}$ be a model of $\mathcal{O}$ and define the set $T_\mathcal{I}$ of all types that are realized in $\mathcal{I}$, that is,

$$T_\mathcal{I} := \{t_\mathcal{I}(d) \mid d \in \Delta^\mathcal{I}\}.$$

Given that $\mathcal{I}$ is a model of $\mathcal{O}$, every element of $T_\mathcal{I}$ is a type according to Definition 2. Using the elements of $T_\mathcal{I}$, we construct a set $\mathbb{T}_\mathcal{I} := \{\mathfrak{t}_\mathcal{I}(d) \mid d \in \Delta^\mathcal{I}\}$ of augmented types. For any domain element $d \in \Delta^\mathcal{I}$, we build the augmented type $\mathfrak{t}_\mathcal{I}(d) = (t_0, \dots, t_{n_\mathcal{O}}, \mathfrak{C}_d)$ as follows.

First, we set $t_0 := t_\mathcal{I}(d) \in T_\mathcal{I}$. Assuming that $\exists r_i.C_i \in \mathcal{M}$ for $i = 1, \dots, n_{\mathsf{ex}}$, we select types $t_1, \dots, t_{n_{\mathsf{ex}}}$ to add to $\mathfrak{t}$ that realize the (possibly negated) existential role restrictions occurring in t_0. If $\exists r_i.C_i \in t_0$, then we can select t_i as the type of an r-successor d' of d such that $d' \in C_i^\mathcal{I}$; otherwise, we pick t_i as the type of an arbitrary element in $\Delta^\mathcal{I}$.

Next, we assume that $\exists p_1^i, \dots, p_k^i.P_i \in \mathcal{M}$ for $i = 1, \dots, n_{\mathsf{cd}}$ and define the function $\mathsf{off}(i,j) := n_{\mathsf{ex}} + (i-1) \cdot n_{\mathsf{ar}} + j$ to be able to refer to the j-th path in the i-th CD-restriction above. We select types $t_{\mathsf{off}(i,1)}, \dots, t_{\mathsf{off}(i,n_{\mathsf{ar}})}$ that realize the (possibly negated) existential CD-restrictions occurring in t_0 for $i = 1, \dots, n_{\mathsf{cd}}$. If $\exists p_1^i, \dots, p_k^i.P_i \in t_0$, then there exist values $v_j^i \in (p_j^i)^\mathcal{I}(d)$ for $j = 1, \dots, k$ such that $(v_1^i, \dots, v_k^i) \in P^D$. If $p_j = rf$ holds for some feature name f and some role name r, let $t_{\mathsf{off}(i,j)}$ be the type of an r-successor d' of d such that $v_j^i = f^\mathcal{I}(d)$. For every $j = 1, \dots, n_{\mathsf{ar}}$ for which $t_{\mathsf{off}(i,j)}$ has not been selected this way, let $t_{\mathsf{off}(i,j)}$ be the type of an arbitrary individual in $\Delta^\mathcal{I}$. Similarly, if $\exists p_1^i, \dots, p_k^i.P_i \notin t_0$ then we set $t_{\mathsf{off}(i,j)}$ be the type of an arbitrary individual in $\Delta^\mathcal{I}$ for $j = 1, \dots, n_{\mathsf{ar}}$.

The two processes described in the two previous paragraphs yield a sequence of types $t_1, \dots, t_{n_\mathcal{O}}$ that occur in $T_\mathcal{I}$ and can thus be associated to individuals $d_1, \dots, d_{n_\mathcal{O}} \in \Delta^\mathcal{I}$. Using these individuals, we define the local system associated to our augmented type $\mathfrak{t}_\mathcal{I}(d)$. First, we define the constraint system $\mathfrak{C}_d$ that contains the constraint $P(f_1^{i_1}, \dots, f_k^{i_k})$ iff $(f_1^\mathcal{I}(d_{i_1}), \dots, f_k^\mathcal{I}(d_{i_k})) \in P^D$ for all $i_1, \dots, i_k \in \{0, \dots, n_\mathcal{O}\}$. We associate to this constraint system a successor function succ_d that assigns to $r \in \mathsf{N_R}$ all $i \in \{1, \dots, n_\mathcal{O}\}$ for which d_i is an r-successor of d. This concludes our definition of $\mathfrak{t}_\mathcal{I}(d)$ for $d \in \Delta^\mathcal{I}$, and thus of $\mathbb{T}_\mathcal{I}$.

Claim 3. *Every augmented type in $\mathbb{T}_\mathcal{I}$ is locally realizable and patched in $\mathbb{T}_\mathcal{I}$.*

Therefore, no augmented type $\mathfrak{t} \in \mathbb{T}_\mathcal{I}$ is eliminated during a run of Algorithm 1, and so $\mathbb{T}_\mathcal{I} \subseteq \mathbb{T}$. We further deduce that $\mathbb{T}$ cannot become empty, since $T_\mathcal{I}$ is non-empty. Using $\mathbb{T}_\mathcal{I}$ together with our model $\mathcal{I}$ of $\mathcal{O}$ we derive an ABox type $\mathfrak{t}_\mathcal{A}^\mathcal{I} = \big((\mathfrak{t}_a)_{a \in \mathsf{N_I}(\mathcal{A})}, \mathcal{A}_R, \mathfrak{C}_\mathcal{A}\big)$ for $\mathcal{O}$. We define each $\mathfrak{t}_a$, $a \in \mathsf{N_I}(\mathcal{A})$, as $\mathfrak{t}_a := \mathfrak{t}_\mathcal{I}(a^\mathcal{I}) \in \mathbb{T}_\mathcal{I}$. Assuming that $d_{a,i}$ is the domain element used to establish the i-th type in $\mathfrak{t}_a$ for $i \in \{0, \dots, n_\mathcal{O}\}$, we define the constraint system $\mathfrak{C}_\mathcal{A}$ s.t.

$$P(f_1^{a_1,i_1}, \dots, f_k^{a_k,i_k}) \in \mathfrak{C}_\mathcal{A} \text{ iff } (f_1^\mathcal{I}(d_{a_1,i_1}), \dots, f_k^\mathcal{I}(d_{a_k,i_k})) \in P^D.$$

Finally, we define the set $\mathcal{A}_R$ to consist of all role assertions $r(a,b)$ for which $a, b \in \mathsf{N_I}(\mathcal{A})$, $r \in \mathsf{N_R}(\mathcal{O})$, and $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$.

Claim 4. *The object $\mathfrak{t}_\mathcal{A}^\mathcal{I}$ is an ABox type for $\mathcal{O}$ with $\mathfrak{t}_a \in \mathbb{T}_\mathcal{I}$ for all $a \in \mathsf{N_I}(\mathcal{A})$.*

Thus, there is a suitable ABox type for $\mathcal{O}$, and Algorithm 1 returns CONSISTENT. □

We can show in a standard way that Algorithm 1 runs in exponential time w.r.t. $\mathcal{O}$ and obtain the following theorem (a detailed proof can be found in [12]).

Theorem 8. *Let $\mathfrak{D}$ be an* ExpTime*-ω-admissible concrete domain. Then, the consistency problem for $\mathcal{ALC}(\mathfrak{D})$ ontologies is* ExpTime*-complete.*

4. Concrete Domain Assertions

Beside using concrete domain restrictions on the concept level, we may want to use feature and predicate assertions in the ABox to either assign a specific value of a feature to some individual or directly constraint the values of features of different individuals. Formally, a *feature assertion* is of the form $f(a, c)$, where $f \in \mathsf{N_F}$, $a \in \mathsf{N_I}$, and $c \in D$, and it is satisfied by an interpretation $\mathcal{I}$ if $f^{\mathcal{I}}(a^{\mathcal{I}}) = c$. *Predicate assertions* are of the form $P\big(f_1(a_1), \dots, f_k(a_k)\big)$, where P is a k-ary relation over $\mathfrak{D}$ and $f_i \in \mathsf{N_F}$, $a_i \in \mathsf{N_I}$ for $i = 1, \dots, k$. An interpretation $\mathcal{I}$ satisfies such an assertion if $\big(f_1^{\mathcal{I}}(a_1^{\mathcal{I}}), \dots, f_k^{\mathcal{I}}(a_k^{\mathcal{I}})\big) \in P^D$. In our setting, we can simulate predicate assertions using concept and role assertions, which leads to the following result.

Theorem 9. *For any* ExpTime*-ω-admissible concrete domain $\mathfrak{D}$, ontology consistency in $\mathcal{ALC}(\mathfrak{D})$ with predicate assertions is* ExpTime*-complete.*

Proof. First, we show how to simulate CD-restrictions of the form $\exists f.\top_{\mathfrak{D}}$ that describe all individuals d in $\mathcal{I}$ for which $f^{\mathcal{I}}(d)$ is defined. Although $\top_{\mathfrak{D}}$ may not be a predicate of $\mathfrak{D}$, by JEPD and the fact that the signature of $\mathfrak{D}$ is non-empty and finite, $\top_{\mathfrak{D}}$ can be expressed as the disjunction of some k-ary predicates $P_1, \dots, P_m$. This implies that for every $d \in D$ there is exactly one k-ary predicate P_i such that $(d, \dots, d) \in P_i^D$. Thus, we can write $\exists f.\top_{\mathfrak{D}}$ equivalently as $\exists f, \dots, f.P_1 \sqcup \dots \sqcup \exists f, \dots, f.P_m$, where each restriction $\exists f, \dots, f.P_i$ repeats f for k times.

Let now $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ be an $\mathcal{ALC}(\mathfrak{D})$ ontology with predicate assertions. We introduce a fresh individual name a^* and fresh role names r_a for all individual names $a \in \mathsf{N_I}(\mathcal{O})$. The ontology $\mathcal{O}'$ is then obtained from $\mathcal{O}$ by adding the role assertions $r_a(a^*, a)$ for all $a \in \mathsf{N_I}(\mathcal{O})$, and replacing all predicate assertions $P\big(f_1(a_1), \dots, f_k(a_k)\big)$ in $\mathcal{A}$ by the concept assertions $(\exists f_1.\top_{\mathfrak{D}})(a_1)$, …, $(\exists f_k.\top_{\mathfrak{D}})(a_k)$, and $(\forall r_{a_1} f_1, \dots, r_{a_k} f_k.P)(a^*)$. Any model $\mathcal{I}$ of $\mathcal{O}'$ satisfies $(c_1, \dots, c_k) \in P^D$ for all possible values $c_i \in (r_{a_i} f_i)^{\mathcal{I}}$, and thus in particular for $c_i = f_i^{\mathcal{I}}(a_i^{\mathcal{I}})$, which shows that $\mathcal{I}$ is also a model of $\mathcal{O}$. Conversely, from every model $\mathcal{I}$ of $\mathcal{O}$ we obtain a model of $\mathcal{O}'$ by choosing an arbitrary element $d^* \in \Delta^{\mathcal{I}}$ for the interpretation of a^* and adding $(d^*, a^{\mathcal{I}})$ to the interpretation of r_a for every individual name a. □

Feature assertions can also be simulated under certain conditions. For concrete domains that contain *singleton* predicates $=_c$ with $(=_c)^D = \{c\} \subseteq D$, we can express any feature assertion $f(a, c)$ using the concept assertion $(\exists f.{=_c})(a)$. However, due to its finite signature, $\mathfrak{D}$ can only contain finitely many such predicates, and hence the feature assertions are restricted by the chosen concrete domain. Due to the JD and JEPD conditions, it turns out that adding feature assertions is actually *equivalent* to adding singleton predicates in the following sense. Here, an *additional singleton predicate* $=_c$ is one that is not part of $\mathfrak{D}$, but otherwise can be used in an ontology with the same semantics as defined above; the proof can be found in [12].

Theorem 10. *For an ω-admissible concrete domain $\mathfrak{D}$, ontology consistency in $\mathcal{ALC}(\mathfrak{D})$ with additional singleton predicates can be polynomially reduced to ontology consistency in $\mathcal{ALC}(\mathfrak{D})$ with feature assertions.*

If we additionally require $\mathfrak{D}$ to be homogeneous, then we can show that arbitrary feature assertions can already be expressed in ordinary $\mathcal{ALC}(\mathfrak{D})$ ontologies.

Theorem 11. *For an* ExpTime*-ω-admissible homogeneous concrete domain $\mathfrak{D}$, ontology consistency in $\mathcal{ALC}(\mathfrak{D})$ with feature assertions is* ExpTime*-complete.*

Proof. Due to Theorem 9, it suffices to provide a reduction to ontology consistency with predicate assertions. Let $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ be an $\mathcal{ALC}(\mathfrak{D})$ ontology containing feature assertions. Let $\mathcal{A}'$ be the ABox containing all concept and role assertions from $\mathcal{A}$ and, in addition, all predicate assertions $P(f_1(a_1), \dots, f_k(a_k)) \in \mathcal{A}'$ for all combinations of feature assertions $f_i(a_i, c_i) \in \mathcal{A}$ with $i = 1, \dots, k$ for which $(c_1, \dots, c_k) \in P^D$ holds. The size of $\mathcal{A}'$ is polynomial in the input, since the signature of $\mathfrak{D}$ is fixed.

It is easy to see that every model of $\mathcal{O}$ is also a model of $\mathcal{O}' := \mathcal{A}' \cup \mathcal{T}$. Conversely, let $\mathcal{I}$ be a model of $\mathcal{O}'$ and let $\mathfrak{D}_{\mathcal{A}}$, $\mathfrak{D}_{\mathcal{I}}$ be the finite substructures of $\mathfrak{D}$ over the domains

$$D_{\mathcal{A}} := \{c \mid f(a,c) \in \mathcal{A}\} \text{ and } D_{\mathcal{I}} := \{f^{\mathcal{I}}(a) \mid f(a,c) \in \mathcal{A}\},$$

respectively. By definition of $\mathcal{A}'$ and JEPD, we know that $\left(f_1^{\mathcal{I}}(a_1^{\mathcal{I}}), \ldots, f_k^{\mathcal{I}}(a_k^{\mathcal{I}})\right) \in P^D$ iff $(c_1, \ldots, c_k) \in P^D$, for all combinations of feature assertions $f_i(a_i, c_i)$ in $\mathcal{A}$. By JD, this in particular implies that $f_1^{\mathcal{I}}(a_1^{\mathcal{I}}) = f_2^{\mathcal{I}}(a_2^{\mathcal{I}})$ iff $f_1(a_1, c), f_2(a_2, c) \in \mathcal{A}$ for some value $c \in D$, which means that the two substructures have the same number of elements. Moreover, by the first equivalence, the mapping $f^{\mathcal{I}}(a^{\mathcal{I}}) \mapsto c$ for all $f(a,c) \in \mathcal{A}$ is an isomorphism between $\mathfrak{D}_{\mathcal{I}}$ and $\mathfrak{D}_{\mathcal{A}}$. Since $\mathfrak{D}$ is homogeneous, there exists an isomorphism $h\colon D \to D$ such that $h(f^{\mathcal{I}}(a^{\mathcal{I}})) = c$ if $f(a,c) \in \mathcal{A}$. Consequently, we define $\mathcal{I}'$ from $\mathcal{I}$ by changing the interpretation of feature names to $f^{\mathcal{I}'}(d) := h(f^{\mathcal{I}}(d))$ iff this value is defined for $f \in \mathsf{N_F}$ and $d \in \Delta^{\mathcal{I}}$. Since h is an isomorphism, we have $C^{\mathcal{I}} = C^{\mathcal{I}'}$ for all concepts C, including CD restrictions, which shows that $\mathcal{I}'$ is a model of $\mathcal{T}$ and all concept and role assertions in $\mathcal{A}$. Moreover, it also satisfies all feature assertions $f(a,c) \in \mathcal{A}$ since $f^{\mathcal{I}'}(a^{\mathcal{I}'}) = h(f^{\mathcal{I}}(a^{\mathcal{I}})) = c$ by construction. □

Together with Theorem 10, this also shows that one can use arbitrary singleton equality predicates in $\mathcal{ALC}(\mathfrak{D})$ ontologies over a homogeneous concrete domain $\mathfrak{D}$, even if $\mathfrak{D}$ contains only finitely many singleton predicates (or none).

5. Conclusion

In this paper, we revisited the problem of reasoning in $\mathcal{ALC}(\mathfrak{D})$ with an ω-admissible concrete domain $\mathfrak{D}$, first addressed in [5]. There, it was conjectured that concept satisfiability w.r.t. a TBox is ExpTime-complete, provided that $\mathrm{CSP}(\mathfrak{D})$ is decidable in exponential time. Using an approach based on type elimination, we successfully proved this conjecture. In addition, we integrated ABox reasoning and showed that reasoning w.r.t. an $\mathcal{ALC}(\mathfrak{D})$ ontology where one can refer to specific values via feature assertions is also ExpTime-complete, if in addition to the above $\mathfrak{D}$ is an ω-admissible homogeneous structure. The main examples of ω-admissible concrete domains from the literature fulfill this requirement as they are (reducts of) finitely bounded homogeneous structures [9], and so we obtain insights into the complexity of reasoning with extensions of $\mathcal{ALC}$ by concrete domain restrictions ranging over Allen's interval algebra, the region connection calculus RCC8, the rational numbers with ordering and equality, and disjoint combinations of those domains.

By extending the type elimination algorithm proposed in this paper appropriately, we believe that it is possible to show that decidability is preserved in extensions of $\mathcal{ALC}(\mathfrak{D})$ such as $\mathcal{ALCI}(\mathfrak{D})$, where inverse roles are allowed in both role and concrete domain restrictions (so that we can write, e.g. $\forall \mathsf{hasChild}^-\, \mathsf{age}, \mathsf{hasChild}\, \mathsf{age}.>$), and $\mathcal{ALCQ}(\mathfrak{D})$, which supports qualified number restrictions. We also plan to use this type elimination approach as a starting point in our investigations of different inference problems, for instance signature-based abduction [14] for $\mathcal{ALC}(\mathfrak{D})$ ontologies and abstract definability for $\mathcal{ALC}(\mathfrak{D})$ TBoxes, i.e. checking whether their *abstract expressive power* [15] can be defined in $\mathcal{ALC}$.

Acknowledgments

This work was partially supported by DFG grant 389792660 as part of TRR 248 – CPEC, by the German Federal Ministry of Education and Research (BMBF, SCADS22B) and the Saxon State Ministry for Science, Culture and Tourism (SMWK) by funding the competence center for Big Data and AI "ScaDS.AI Dresden/Leipzig". The authors would like to thank Jakub Rydval for his help in understanding the properties of finitely bounded homogeneous structures.

References

[1] F. Baader, P. Hanschke, A Scheme for Integrating Concrete Domains into Concept Languages, in: J. Mylopoulos, R. Reiter (Eds.), Proceedings of the 12th International Joint Conference on Artificial

Intelligence. Sydney, Australia, August 24-30, 1991, Morgan Kaufmann, 1991, pp. 452–457. URL: http://ijcai.org/Proceedings/91-1/Papers/070.pdf.

[2] C. Lutz, The complexity of description logics with concrete domains, Ph.D. thesis, RWTH Aachen University, Germany, 2002. URL: http://nbn-resolving.org/urn:nbn:de:hbz:82-opus-3032.

[3] C. Lutz, Description logics with concrete domains - A survey, in: P. Balbiani, N. Suzuki, F. Wolter, M. Zakharyaschev (Eds.), Advances in Modal Logic 4, King's College Publications, 2002, pp. 265–296. URL: http://www.aiml.net/volumes/volume4/Lutz.ps.

[4] F. Baader, S. Brandt, C. Lutz, Pushing the EL envelope, in: L. P. Kaelbling, A. Saffiotti (Eds.), IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Professional Book Center, 2005, pp. 364–369. URL: http://ijcai.org/Proceedings/05/Papers/0372.pdf.

[5] C. Lutz, M. Miličić, A Tableau Algorithm for Description Logics with Concrete Domains and General TBoxes, Journal of Automated Reasoning 38 (2007) 227–259. doi:10.1007/s10817-006-9049-7.

[6] A. Artale, V. Ryzhikov, R. Kontchakov, DL-Lite with attributes and datatypes, in: L. D. Raedt, C. Bessiere, D. Dubois, P. Doherty, P. Frasconi, F. Heintz, P. J. F. Lucas (Eds.), ECAI 2012 - 20th European Conference on Artificial Intelligence, volume 242 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2012, pp. 61–66. doi:10.3233/978-1-61499-098-7-61.

[7] C. Carapelle, A.-Y. Turhan, Description Logics Reasoning w.r.t. General TBoxes Is Decidable for Concrete Domains with the EHD-Property, in: ECAI 2016, IOS Press, 2016, pp. 1440–1448. doi:10.3233/978-1-61499-672-9-1440.

[8] N. Labai, M. Ortiz, M. Šimkus, An ExpTime Upper Bound for ALC with Integers, Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning 17 (2020) 614–623. doi:10.24963/kr.2020/61.

[9] F. Baader, J. Rydval, Using Model Theory to Find Decidable and Tractable Description Logics with Concrete Domains, Journal of Automated Reasoning 66 (2022) 357–407. doi:10.1007/s10817-022-09626-2.

[10] S. Demri, K. Quaas, First Steps Towards Taming Description Logics with Strings, in: S. Gaggl, M. V. Martinez, M. Ortiz (Eds.), Logics in Artificial Intelligence, Lecture Notes in Computer Science, Springer Nature Switzerland, Cham, 2023, pp. 322–337. doi:10.1007/978-3-031-43619-2_23.

[11] F. Baader, J. Rydval, Description Logics with Concrete Domains and General Concept Inclusions Revisited, in: N. Peltier, V. Sofronie-Stokkermans (Eds.), Automated Reasoning, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2020, pp. 413–431. doi:10.1007/978-3-030-51074-9_24.

[12] S. Borgwardt, F. De Bortoli, P. Koopmann, The Precise Complexity of Reasoning in $\mathcal{ALC}$ with ω-Admissible Concrete Domains (Extended Version), 2024. arXiv:2405.19096.

[13] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An Introduction to Description Logic, Cambridge University Press, 2017. doi:10.1017/9781139025355.

[14] P. Koopmann, Signature-based abduction with fresh individuals and complex concepts for description logics, in: Z. Zhou (Ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI, ijcai.org, 2021, pp. 1929–1935. doi:10.24963/IJCAI.2021/266.

[15] F. Baader, F. De Bortoli, The Abstract Expressive Power of First-Order and Description Logics with Concrete Domains, in: Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, SAC '24, ACM, New York, NY, USA, 2024, pp. 754–761. doi:10.1145/3605098.3635984.