ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1440

Description Logics Reasoning w.r.t. General TBoxes Is Decidable for Concrete Domains with the EHD-Property

Claudia Carapelle and **Anni-Yasmin Turhan**[1]

Abstract. Reasoning for Description logics with concrete domains and w.r.t. general TBoxes easily becomes undecidable. However, with some restriction on the concrete domain, decidability can be regained. We introduce a novel way to integrate concrete domains $\mathcal{D}$ into the well-known description logic $\mathcal{ALC}$, we call the resulting logic $\mathcal{ALC}^P(\mathcal{D})$. We then identify sufficient conditions on $\mathcal{D}$ that guarantee decidability of the satisfiability problem, even in the presence of general TBoxes. In particular, we show decidability of $\mathcal{ALC}^P(\mathcal{D})$ for several domains over the integers, for which decidability was open. More generally, this result holds for all negation-closed concrete domains with the EHD-*property*, which stands for 'the existence of a homomorphism is definable'. Such technique has recently been used to show decidability of CTL* with local constraints over the integers.

1 Introduction

Description Logics (DLs) are a collection of knowledge representation formalisms with well-founded semantics. Most DLs are (decidable) fragments of First Order Logic (FO). They are employed nowadays in a range of application areas such as the bio-medical field or the semantic web, and are the foundations of the web ontology language OWL 2 [20]. DLs are an excellent tool to represent abstract knowledge and to reason over it, but practical applications often require concrete properties with values from a fixed domain, such as integers or strings and to support built-in predicates.

In [1], DLs were extended with *concrete domains*, where partial functions map objects of the abstract domain to values of the concrete domain. The resulting logic $\mathcal{ALC}(\mathcal{D})$ extends the standard DL $\mathcal{ALC}$ by *concrete domain restrictions* over a concrete domain $\mathcal{D}$. Concrete domain restrictions can be used for building complex concepts based on concrete qualities of their instances such as the age, temperature or even measured values. For instance, the following GCI of $\mathcal{ALC}(\mathcal{D})$-concepts:

$$\mathsf{motor\text{-}vehicle\text{-}driver} \sqsubseteq \mathsf{Person} \sqcap \exists \mathsf{has\text{-}age}.\geq_{18}$$

requires that drivers of a motor vehicle are at least 18 years old. A concrete domain restriction can connect several abstract objects via *feature-paths*, i.e. paths of functional roles, and assert a predicate of arbitrary arity for concrete quantities of those objects. Concrete domains are incorporated in a weakened form in OWL as data-types for which only unary predicates are admitted [20].

If definitorial, acyclic TBoxes are used, then reasoning for $\mathcal{ALC}$ extended by concrete domains that are *admissible* is decidable [1]. Reasoning can become undecidable in the presence of general TBoxes [11, 15] for such DLs. There have been several attempts to regain decidability for reasoning in $\mathcal{ALC}(\mathcal{D})$ with general TBoxes. Some approaches simply restrict concrete domain restrictions to unary predicates [10] or to feature-paths of length 1 [9]. These restrictions limit the modelling capabilities severely. Lutz and Miličić took a different approach and showed that if a concrete domain respects a criterion called ω-*admissibility*, then satisfiability for $\mathcal{ALC}(\mathcal{D})$ with general TBoxes is decidable [16]. The condition of ω-admissibility essentially allows to lift local satisfiability of (connected) concrete domain parts to global satisfiability by requiring compactness and that the concrete domain parts need to conform on the predicates asserted for the shared objects. This condition indicates decidability of DL reasoning for some concrete domains, for instance, the RCC8 relations and the Allen relations over the real numbers [16]. However, several interesting domains do not satisfy ω-admissibility, for instance, the ones based on non-dense numerical sets, as the integers or the natural numbers. In [14] Lutz considers a concrete domain over the rational numbers, and proves that reasoning w.r.t. general TBoxes is decidable. Such domain can, however, not be used to reasonably represent some situations: certain concrete features, such as 'number of children', cannot possibly be fractions.

In this paper we devise a new criterion for concrete domains that guarantees decidability of the satisfiability problem in the presence of general TBoxes. This criterion holds also for some concrete domains that are known to be not ω-admissible, such as the integers. To this end we introduce the new DL $\mathcal{ALC}^P(\mathcal{D})$ that uses path constraints instead of concrete domain restrictions. Unlike the latter, which only allow feature-paths to connect an individual and a concrete value, path constraints can use the full expressiveness of role-paths. This enables to model for instance 'person who only has younger siblings', as an individual whose age is greater than that of all his siblings, where the sibling relation need not be functional. Furthermore, $\mathcal{ALC}^P(\mathcal{D})$ admits Boolean combinations of concrete domain predicates in path constraints.

We show decidability of the satisfiability problem of $\mathcal{ALC}^P(\mathcal{D})$-concepts w.r.t. general TBoxes if $\mathcal{D}$ (1) is *negation-closed*, which requires that the complement of each (atomic) relation is effectively definable by a positive existential first-order formula, and (2) has the EHD-*property*, which stands for 'the existence of a homomorphism is definable', expressing the ability of a certain logic L to distinguish between those structures which can be mapped to $\mathcal{D}$ by a homomorphism and those who cannot. Our approach to show decidability of $\mathcal{ALC}^P(\mathcal{D})$ with concrete domains that fulfill the above conditions is an adaptation of the EHD-method, used in [6, 7] for CTL* and ECTL*. This, in turn, uses a recent decidability result by Bojańczyk and Toruńczyk for WMSO+B over infinite trees, an extension of

[1] Technische Universität Dresden, Institute for Theoretical Computer Science, SFB HAEC; email: firstname.lastname@tu-dresden.de

weak monadic second order logic by the bounding quantifier B ([3]).

The idea for testing satisfiability of an $\mathcal{ALC}^P(\mathcal{D})$-concept C w.r.t. an $\mathcal{ALC}^P(\mathcal{D})$-TBox $\mathcal{T}$ is to proceed in two steps. First, an ordinary $\mathcal{ALC}$ interpretation is built that satisfies an abstracted version of C and $\mathcal{T}$, where each path constraint is replaced by a fresh concept name. Second, this interpretation is used to generate a so-called constraint graph, which is a structure for storing the contribution of the constraints that were abstracted away. We show that deciding whether such a constraint graph allows a homomorphism to the concrete domain is enough to guarantee that the constraints are satisfied. In contrast to the mentioned CTL variants, $\mathcal{ALC}^P(\mathcal{D})$ is multi-modal and uses features, i.e. functional roles, which required some adaptation to apply the techniques from [6, 7].

By the newly established criterion for decidability of $\mathcal{ALC}^P(\mathcal{D})$ w.r.t. TBoxes, we confirm what the authors of Lutz and Miličić have conjectured: that ω-admissibility is a sufficient, but not necessary condition for decidability. We show, in fact, that reasoning with non ω-admissible concrete domains over the natural numbers and the integers w.r.t. general TBoxes is decidable. We also show that it is possible to consider an extension of the concrete domain over the rational numbers presented in [12, 11], that allows to test whether a certain concrete value is an integer.

This paper is structured as follows. In the next two sections we give some preliminary notions and introduce the DL $\mathcal{ALC}^P(\mathcal{D})$ with some of its properties. Section 4 explains the EHD-method and shows decidability of DL reasoning for $\mathcal{ALC}^P(\mathcal{D})$ with negation-closed concrete domains that have the EHD-property. As customary, the paper ends with conclusions and future work.
All omitted or shortened proofs can be found in full detail in [8].

2 Preliminaries

Before we define our DL $\mathcal{ALC}^P(\mathcal{D})$, we introduce some basic notions needed later on in the technical constructions. A (*relational*) *signature* $\sigma = \{R_1, R_2, \ldots\}$ is a countable (finite or infinite) set of relation symbols. Every relation symbol $R \in \sigma$ has an associated arity $\mathsf{ar}(R) \geq 1$. A *σ-structure* is a tuple $\mathcal{A} = (A, R_1^{\mathcal{A}}, R_2^{\mathcal{A}}, \ldots)$, where A is a non-empty set and for each $R \in \sigma$, $R^{\mathcal{A}} \subseteq A^{\mathsf{ar}(R)}$ is the interpretation of the relation symbol R in $\mathcal{A}$, that is an $\mathsf{ar}(R)$-ary relation over A.

Example 1. A simple example of a $\{=, <\}$-structure is $\mathcal{Z} = (\mathbb{Z}, =^{\mathcal{Z}}, <^{\mathcal{Z}})$, where $=^{\mathcal{Z}}$ and $<^{\mathcal{Z}}$ are defined as expected, namely as $\{(a, b) \in \mathbb{Z}^2 \mid a = b\}$ and $\{(a, b) \in \mathbb{Z}^2 \mid a < b\}$, respectively.

We often identify the relation $R^{\mathcal{A}}$ with the relation symbol R. In the example above, then, we would simply write $(\mathbb{Z}, =, <)$. For a σ-structure $\mathcal{A}$ and a τ-structure $\mathcal{B}$ such that $\tau \subseteq \sigma$, a *homomorphism* from $\mathcal{B}$ to $\mathcal{A}$ is a mapping $h : B \to A$ such that for all $R \in \tau$ and all tuples $(b_1, \ldots, b_{\mathsf{ar}(R)}) \in B^{\mathsf{ar}(R)}$ we have

$$(b_1, \ldots, b_{\mathsf{ar}(R)}) \in R^{\mathcal{B}} \Rightarrow (h(b_1), \ldots, h(b_{\mathsf{ar}(R)})) \in R^{\mathcal{A}} .$$

We write $\mathcal{B} \preceq \mathcal{A}$ if there is a homomorphism from $\mathcal{B}$ to $\mathcal{A}$. Note that we do not require this homomorphism to be injective.

We shortly introduce MSO and WMSO+B, for a more detailed introduction we refer the reader to [3, 18]. We fix countably infinite sets $\mathsf{V_e}$ and $\mathsf{V_s}$ of element variables and set variables, respectively. *Monadic second-order logic* (MSO) is the extension of first-order logic (FO) where also quantification over sets is allowed. MSO-formulas over a signature σ are defined by the following grammar, where $R \in \sigma, x, y, x_1, \ldots, x_{\mathsf{ar}(R)} \in \mathsf{V_e}$ and $X \in \mathsf{V_s}$:

$$\varphi := R(x_1, \ldots, x_{\mathsf{ar}(R)}) \,|\, x = y \,|\, x \in X \mid \neg\varphi \,|\, (\varphi \wedge \varphi) \,|\, \exists x\varphi \,|\, \exists X\varphi.$$

Using negation we can obtain disjunction $\vee$, universal quantification $\forall x$ and $\forall X$, and implication $\to$. MSO-formulas are evaluated on σ-structures, where element and set variables range respectively over elements and subsets of the domain. *Weak monadic second-order logic* (WMSO) has the same syntax as MSO, but second-order variables are interpreted as finite subsets of the underlying universe.

WMSO+B is the extension of WMSO by the *bounding quantifier* $\mathsf{B}X\varphi$ for $X \in \mathsf{V_s}$. The semantics of $\mathsf{B}X\varphi$ on a structure $\mathcal{A}$ with universe A is defined as follows: $\mathcal{A} \models \mathsf{B}X\varphi(X)$ if and only if there is a bound $b \in \mathbb{N}$ such that whenever $\mathcal{A} \models \varphi(B)$ for some finite subset $B \subseteq A$, then $|B| \leq b$.

Finally, let BMWB denote the set of all Boolean combinations of MSO-formulas and (WMSO+B)-formulas.

Example 2. Given a graph $\mathcal{G} = (V, E)$, WMSO can express reachability in $\mathcal{G}$. We define the WMSO-formula $\mathsf{reach}(x_1, x_2)$ to be

$$\exists Z\, x_1 \in Z \wedge \forall Y \subseteq Z\big[\big(x_1 \in Y \wedge \mathsf{scl}(Y)\big) \to x_2 \in Y\big] ,$$

where $\mathsf{scl}(Y) = \forall y \forall z(y \in Y \wedge z \in Z \wedge E(y, z)) \to z \in Y$ says that the set Y is successor-closed. The semantics of reach seen as an MSO-formula or a WMSO-formula are the same because b is reachable from a in the graph $\mathcal{G}$ if and only if it is in some finite subgraph of $\mathcal{G}$.

In [3] Bojańczyk and Toruńczyk show that satisfiability for WMSO+B over binary trees is decidable. This result can be extended to BMWB over trees of branching degree n (n-trees):

Theorem 3 (cf. [3, 7])**.** *One can decide whether for a given formula $\varphi \in$ BMWB there exists an n-tree $\mathcal{T}_n$ such that $\mathcal{T}_n \models \varphi$.*

3 The Description Logic $\mathcal{ALC}^P(\mathcal{D})$

We introduce now the new DL $\mathcal{ALC}^P(\mathcal{D})$ and some basic notions on DLs in general. We start with the (concrete domain) constraints.

Let us fix for the rest of this section a countably infinite set of register variables Reg, a relational signature σ, and an arbitrary σ-structure $\mathcal{D} = (D, R_1, R_2, \ldots)$, called the *concrete domain*.

Definition 4. We define a *constraint* $c(x_1, \ldots, x_k)$ of arity k over $\mathcal{D}$ as a Boolean combination of *atomic constraints* $R(x_{i_1}, \ldots, x_{i_{\mathsf{ar}(R)}})$, where $R \in \sigma$ and $i_j \in \{1, \ldots, k\}$. We write $\mathcal{D} \models c(a_1, \ldots, a_k)$ if the constraint is satisfied in $\mathcal{D}$ by the assignment $x_i \mapsto a_i$.

Example 5. Consider as concrete domain $\mathcal{Z} = (\mathbb{Z}, <, =)$, the relational structure introduced in Example 1. Using infix notation for the relations, $c(x, y, z) = [(x < y \vee x = y) \wedge \neg\, y < z]$ is a constraint of arity 3 over $\mathcal{Z}$, and $\mathcal{Z} \models c(0, 1, 0)$.

Let us fix two countably infinite sets $\mathsf{N_C}$ and $\mathsf{N_R}$ of *concept names* and *role names* respectively. Let then $\mathsf{N_F} \subseteq \mathsf{N_R}$ be the set of *features*, i.e. roles that are interpreted as partial functions. We call a finite sequence $P = r_1 \cdots r_n$ of role names a *role-path* of length n.

Definition 6. We recursively define $\mathcal{ALC}^P(\mathcal{D})$*-concepts* as follows

$$C := A \mid \neg C \mid (C \sqcap C) \mid \exists r.C \mid \exists P.c(S^{i_1}x_1, \ldots, S^{i_k}x_k)$$

where $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, P is a role-path of length $n \geq 0$, c is a constraint of arity k, $x_1, \ldots, x_k \in$ Reg, and $i_1, \ldots, i_k \leq n$. We call $\exists P.c(S^{i_1}x_1, \ldots, S^{i_k}x_k)$ a *path constraint*. The symbol S appearing in the path constraints stands for successor, as the term $S^i x$ points at the register variable x in the i-th position of the path P.

$\mathcal{ALC}$ is the fragment of $\mathcal{ALC}^P(\mathcal{D})$ without path constraints. As usual, a *general concept inclusion* (GCI) is an expression of the form $C \sqsubseteq D$, where C and D are concepts. A *TBox* is a finite set of GCIs.

Definition 7. A $\mathcal{D}$-*interpretation* $\mathcal{I}$ is a tuple $(\Delta, \cdot^{\mathcal{I}}, \gamma)$, where Δ is a set called *the domain*, $\cdot^{\mathcal{I}}$ is the *interpretation function*, and $\gamma : \Delta \times \mathsf{Reg} \to D$ is the *valuation function*, assigning a value from the concrete domain to each register variable in each element of the interpretation domain. The interpretation function maps each concept name $A \in \mathsf{N_C}$ to some $A^{\mathcal{I}} \subseteq \Delta$, each role name $r \in \mathsf{N_R}$ to a binary relation $r^{\mathcal{I}} \subseteq \Delta \times \Delta$, with the condition that if $r = f \in \mathsf{N_F}$ the binary relation $f^{\mathcal{I}}$ has to be functional, i.e. for all $a, b, c \in \Delta$, $(a, b), (a, c) \in f^{\mathcal{I}}$ implies $b = c$. It is then extended to $\neg C, C \sqcap D, \exists r.D$ as usual, and to a role-path $P = (r_1, \ldots, r_n)$ as:

$$P^{\mathcal{I}} := \{(v_0, \ldots, v_n) \in \Delta^{n+1} \mid (v_{i-1}, v_i) \in r_i^{\mathcal{I}} \text{ for } i = 1, \ldots, n\}.$$

Finally, if P has length n, we define $(\exists P.c(S^{i_1}x_1, \ldots, S^{i_k}x_k))^{\mathcal{I}}$ as

$$\{v \in \Delta \mid \exists (v_0, \ldots, v_n) \in P^{\mathcal{I}} \text{ s.t. } v_0 = v, \text{ and } \mathcal{D} \models c(\gamma(v_{i_1}, x_1), \ldots, \gamma(v_{i_k}, x_k))\}.$$

So the fact that an element $v \in \Delta$ belongs to the interpretation of a path constraint $\exists P.c(S^{i_1}x_1, \ldots, S^{i_k}x_k)$ means that there exists an instance of the path $P^{\mathcal{I}}$ starting in v, namely some $(v_0, v_1, \ldots, v_n) \in P^{\mathcal{I}}$ with $v_0 = v$, such that the assignment $y_j \mapsto \gamma(v_{j_i}, x_j)$ satisfies the constraint $c(y_1, \ldots, y_k)$. A term S^i inside the constraint is used to point at the i-th element of the path $P^{\mathcal{I}}$. Note that the requirement that $i_1, \ldots, i_k \leq n$ ensures that such element is well-defined.

Note, also that an atomic constraint $R(S^{i_1}x_1, \ldots, S^{i_k}x_k)$ is *local* in the sense that it involves only nodes in a fixed *neighborhood* of the position at which they are evaluated. We call $d := \max\{i_1, \ldots, i_k\}$ the *depth* of R. By extension, the depth of a constraint c is the maximum depth of all the atomic constraints which appear in c.

Let $\mathsf{Reg}_{C,\mathcal{T}}$ denote the set of register variables that occur in C and $\mathcal{T}$. Obviously, the relevance of the valuation function γ is limited to the domain $(\Delta \times \mathsf{Reg}_{C,\mathcal{T}})$.

Definition 8. A $\mathcal{D}$-interpretation $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$ ($\mathcal{I} \models \mathcal{T}$) if and only if every GCI $C \sqsubseteq D \in \mathcal{T}$ is satisfied, that is, if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Given a concept C and a TBox $\mathcal{T}$, we say C is *satisfiable with respect to* $\mathcal{T}$ if and only if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$. We write $\mathcal{I} \models_{\mathcal{T}} C$.

We define some usual abbreviations: $C \sqcup D := \neg(\neg C \sqcap \neg D)$, $\forall r.C := \neg\exists r.\neg C$, $\forall P.c := \neg\exists P.\neg c$, $\exists P.C := \exists r_1.\exists r_2.\cdots\exists r_n.C$, where $P = r_1 \cdots r_n$, and special concept $\top := A \sqcup \neg A$

Using this extended set of operators and DeMorgan's laws we can, given an $\mathcal{ALC}^P(\mathcal{D})$-concept C, obtain an equivalent concept in negation normal form $\mathsf{nnf}(C)$, where negation only appears before concept names or atomic constraints.

The *TBox-concept* of a TBox $\mathcal{T}$ is $C_{\mathcal{T}} := \bigsqcap_{C \sqsubseteq D \in \mathcal{T}} (\neg C \sqcup D)$. Note that it is equivalent to ask that an interpretation $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}}, \gamma_{\mathcal{I}})$ satisfies all GCIs from $\mathcal{T}$ and to ask that the $C_{\mathcal{T}}$ is globally satisfied, i.e. $(C_{\mathcal{T}})^{\mathcal{I}} = \Delta$. Vice-versa, any globally satisfied concept C can be seen as the GCI $\top \sqsubseteq C$. For technical reasons, it is convenient for us to adopt this view, and from now on we will always assume that a TBox consists of a single concept $C_{\mathcal{T}}$ that needs to be globally satisfied. We say a TBox $\mathcal{T}$ is in negation normal form if so is $C_{\mathcal{T}}$.

Example 9. Take again $\mathcal{Z} = (\mathbb{Z}, <, =)$ as concrete domain and consider the following TBox: $\mathcal{T} = \{\exists \mathsf{neighbor}.(\textit{green grass} < S\textit{green grass}),\ \neg\mathsf{GreenThumb} \sqcup (\textit{alive plants} = \textit{plants})\}$[2]. Here we consider three register variables: *green grass* measures the degree of 'greenness' of an individual's lawn, while *plants* and *alive plants* count the number of plants (total or alive) of an individual. In any model of $\mathcal{T}$, every individual has a neighbor whose grass is greener, and individuals with a green thumb keep all their plants alive.

[2] Here the absence of a path quantifier before (*alive plants* = *plants*) means that we are referring to a 'path of length zero'.

In Example 9, there cannot exist a model for $\mathcal{T}$ with a finite underlying domain, as the degree of greenness of neighboring lawns is strictly increasing. This is never the case for ordinary $\mathcal{ALC}$, which enjoys the finite model property.

In the literature on description logics with concrete domains (for instance in [1, 16]) one finds constraints of the kind $\exists R(P_1x_1, \ldots, P_kx_k)$, where R is a relation from the concrete domain and each P_i is a path composed of features only. The constraint is satisfied by an element d if there exist k elements, $d_1 \ldots d_k$, reachable from d via the feature-paths $P_1 \ldots P_k$, such that the tuple $(\gamma(d_1, x_1), \ldots, \gamma(d_k, x_k))$ belongs to the relation R in the concrete domain. Nonetheless, in many interesting cases this kind of constraint can be replaced with path constraints by introducing some additional register variables. For example $\exists(P_1x_1 < P_2x_2)$ can be expressed as $\exists P_1.(S^{|P_2|}x_1 < z) \sqcap \exists P_2.(z \leq S^{|P_2|}x_2)$, where z is a fresh register variable. Also $\forall(P_1x_1 < P_2x_2)$ can be replaced by $\neg(\exists P_1.\top \sqcap \exists P_2.\top) \sqcup (\exists P_1.(S^{|P_1|}x_1 < z) \sqcap \exists P_2.(z \leq S^{|P_2|}x_2))$.[3]

On the other hand, our constraints can use role-paths of arbitrary length, which—to the best of our knowledge—is not allowed in the previously existing literature, where they are limited in length or disallowed completely in favor of feature-paths. Therefore, although generally incomparable in expressiveness, path constraints are strictly more expressive on interesting concrete domains.

Note also that for each individual v of the abstract domain, the value $\gamma(v, x)$ is defined for all $x \in \mathsf{Reg}$. This is essentially the same as saying that each $\gamma(\cdot, x)$, in literature commonly called *concrete feature*, is interpreted as a *total* function, more in the style of the *attributes* used by Toman and Weddell (see [19]).

3.1 $\mathcal{ALC}^P(\mathcal{D})$ has the Tree Model Property

Definition 10. Let $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}}, \gamma)$ be a $\mathcal{D}$-interpretation and define $\to := \bigcup_{r \in \mathsf{N_R}} r^{\mathcal{I}}$. We say $\mathcal{I}$ is a *tree-shaped* $\mathcal{D}$-*interpretation* if and only if $(\Delta, \to)$ is a tree, that is:

- $\Delta \subseteq \Sigma^*$ is (isomorphic to) a prefix-closed set of strings over some alphabet Σ, and
- for all $u, v \in \Delta$, $u \to v$ if and only if $v = ua$ for some $a \in \Sigma$.

We call $\mathcal{I}$ an n-*tree* $\mathcal{D}$-*interpretation* if $\Delta = [1, n]^*$ for some $n \in \mathbb{N}$, where $[1, n]$ denotes the closed interval $\{1, \ldots, n\}$.

A logic has the *tree model property*, if for every concept C and every TBox $\mathcal{T}$, C is satisfiable w.r.t. $\mathcal{T}$ iff there exists a tree-shaped $\mathcal{D}$-interpretation $\mathcal{J}$ such that $\mathcal{J} \models_{\mathcal{T}} C$.

We show that $\mathcal{ALC}^P(\mathcal{D})$ has a strong version of the tree model property, in which we are able to give a bound on the branching degree. Let $\mathcal{T}$ and C be an $\mathcal{ALC}^P(\mathcal{D})$ TBox and concept, respectively. We denote by $\mathsf{Sub}(\mathcal{T}, C)$ the set of all concepts which appear in $\mathcal{T}$ and C. If d is the maximum depth of an existential path constraint occurring in $\mathcal{T}$ and C, and e is the number of existentially quantified subconcepts in $\mathsf{Sub}(\mathcal{T}, C)$ we can prove the following:

Lemma 11. *C is satisfiable w.r.t. $\mathcal{T}$ iff there exists an n-tree $\mathcal{D}$-interpretation $\mathcal{I}$ such that $\mathcal{I} \models_{\mathcal{T}} C$, where $n = d \cdot e$.*

[3] Such translations must be applied after the concepts are converted to strong negation normal form (see Sec. 3.2) because they preserve satisfiability but are not necessarily closed under negation.

Sketch of proof. First we show the normal tree model property, obtained by unraveling. We need to deal with the constraints, but this does not create particular problems. Successively we *prune* the tree, going top bottom and leaving only those nodes that are necessary to satisfy the existentially quantified formulas. These are at most e in each node, and are witnessed at most by a path of d nodes. Once the pruning is finished we have a tree-shaped interpretation $\mathcal{J}$ with branching degree at most n. We can obtain an n-tree: for each node we introduce the needed number of s-successors, where s is a role that does not appear in C or $\mathcal{T}$, and attach a copy of $\mathcal{J}$ to it until we obtain an n-tree. This guarantees that the freshly introduced nodes respect the TBox-concept. □

Given this result, from now on we can restrict ourselves to $\mathcal{D}$-interpretations of the form $\mathcal{I} = ([1,n]^*, \cdot^{\mathcal{I}}, \gamma_{\mathcal{I}})$ where for each $u, v \in [1,n]^*$ there exists $r \in \mathsf{N_R}$ such that $(u,v) \in r^{\mathcal{I}}$ if and only if there exists $i \in [1,n]$ such that $v = ui$.

3.2 Strong Negation Normal Form

We show now how, requiring that the concrete domain satisfies a property called *negation closure*, we can obtain a *strong negation normal form*, where negation only appears in front of concept names.

Definition 12. We call a σ-structure $\mathcal{D} = (D, R_1^{\mathcal{D}}, R_2^{\mathcal{D}}, \dots)$ *negation-closed* if for every $R \in \sigma$ the complement of $R^{\mathcal{D}}$ is effectively definable by a positive existential first-order formula, i.e., if there is a computable function that maps each relation symbol $R \in \sigma$ to a positive existential first-order formula $\varphi_R(x_1, \dots, x_{\mathsf{ar}(R)})$ (a formula that is built up from relations of σ using $\wedge$, $\vee$, and $\exists$) such that

$$D^{\mathsf{ar}(R)} \setminus R^{\mathcal{D}} = \{(a_1, \dots, a_{\mathsf{ar}(R)}) \mid \mathcal{D} \models \varphi_R(a_1, \dots, a_{\mathsf{ar}(R)})\}.$$

Example 13. Let $=_a := \{a\}$ be the unary predicate which holds only for a, and $\equiv_{a,b} := \{a + kb \mid k \in \mathbb{Z}\}$ be a unary predicate expressing that some number is congruent to a modulo b. Consider the structure $(\mathbb{Z}, <, =, (=_a)_{a\in\mathbb{Z}}, (\equiv_{a,b})_{0\le a<b})$. Such a structure is negation-closed, we have in fact:

- $\neg x = y$ if and only if $x < y \ \vee \ y < x$,[4]
- $\neg x < y$ if and only if $x = y \ \vee \ y < x$,
- $\neg x = a$ if and only if $\exists y\,(y = a \wedge (x < y \ \vee \ y < x))$, and
- $\neg x \equiv a \bmod b$ if and only if $x \equiv c \bmod b$ for some $0 \le c < b$ with $a \ne c$:
$$\bigvee_{\substack{0\le c<b\\ a\ne c}} x \equiv c \bmod b\,.$$

Definition 14. We say that an $\mathcal{ALC}^P(\mathcal{D})$-concept φ is in *strong negation normal form* if it is in negation normal form and if, additionally, all constraints $c(x_1, \dots, x_k)$ do not contain any negation. Consequently we say that a TBox $\mathcal{T}$ is in strong negation normal form if so is the TBox-concept $C_{\mathcal{T}}$.

Lemma 15. *If $\mathcal{D} = (D, R_1^{\mathcal{D}}, R_2^{\mathcal{D}}, \dots)$ is negation-closed, given a concept C and a TBox $\mathcal{T}$, one can compute $\widehat{C}$ and $\widehat{\mathcal{T}}$ in strong negation normal form such that C is satisfiable with respect to $\mathcal{T}$ if and only if $\widehat{C}$ is satisfiable with respect to $\widehat{\mathcal{T}}$.*

Sketch of proof. The idea is the following: Given a negated atomic constraint $\neg R(\dots)$ of depth d, we want to substitute it with a boolean combination of positive ones. We use the fact that the complement of R is definable by an existential first order sentence φ_R. We deal with the variables existentially quantified in φ_R by adding fresh register variables. These new registers are 'placed' at depth d, so that (considering the tree-like structure of an $\mathcal{ALC}^P(\mathcal{D})$ model) they are unique for the path used to evaluate R. □

Example 16. Consider the concrete domain $(\mathbb{Z}, <, =, (=_a)_{a\in\mathbb{Z}})$ and the concept $C = \exists rs.[S^1x < S^2x \wedge \neg S^2x = 3]$. An individual v which belongs to an interpretation of C must necessarily have an r-successor v_1 which has an s-successor v_2, such that the value of x in v_1 is smaller than the value of x in v_2, which in turn must be different than 3. As one can see from Example 13, $\mathcal{D}$ is negation-closed, and we can find an existentially quantified positive first order formula, namely $\psi(a) = \exists z(z = 3 \wedge (a < z \vee z < a))$, such that $\neg x = 3$ if and only if $\psi(x)$ holds. The strong negation normal form of C is $\widehat{C} = \exists rs.[S^1x < S^2x \wedge S^2y = 3 \wedge (S^2x < S^2y \vee S^2y < S^2x)]$. As you can see we have introduced a new register variable y and placed it at depth 2 inside the constraint to hold the value that was existentially quantified in ψ.

Now that we have successfully eliminated negation from inside the constraints, there is one last step to do, in order to obtain a normal form that will be useful in the next section. Observe that if a constraint $c(x_1, \dots, x_k)$ does not contain negation, it is possible to apply distributivity repeatedly and obtain an equivalent constraint in DNF or in CNF[5] which still does not contain negation. Therefore we can assume that all path constraints of the form $\exists P.c$ (respectively $\forall P.c$) are such that the constraint c is in DNF (resp. CNF). Using then the fact that universal quantification commutes with conjunction and that existential quantification commutes with disjunction, we can easily prove the following facts:

$$\exists P. \bigvee_{i=1}^{n} (a_1^i \wedge \dots \wedge a_{m_i}^i) \equiv \bigsqcup_{i=1}^{n} \exists P.(a_1^i \wedge \dots \wedge a_{m_i}^i), \text{ and}$$

$$\forall P. \bigwedge_{i=1}^{n} (a_1^i \vee \dots \vee a_{n_i}^i) \equiv \bigsqcap_{i=1}^{n} \forall P.(a_1^i \vee \dots \vee a_{n_i}^i),$$

where each a_j^i is an atomic constraint. Therefore, given a concept C in strong negation normal form, and applying the above described transformations, we can obtain a new concept C' which is still in strong negation normal form, and is such that all path constraints are of the kind $\exists P.c$ (or $\forall P.c$) where c is a conjunction (resp. disjunction) of atomic constraints. We call this the *constraint normal form* of C.

4 The EHD Method

Following the approach in [6, 7] for CTL* and ECTL*, we can reduce the satisfiability problem of $\mathcal{ALC}^P(\mathcal{D})$ to the satisfiability problem for BMWB over n-trees, provided the concrete domain has right properties.

4.1 The EHD-Property

The central notion used in the decidability proof is the EHD-*property*. EHD stands for 'the existence of a homomorphism is definable'. It is a property of a relational structure $\mathcal{A}$, expressing the ability of a logic L to distinguish between those structures $\mathcal{B}$ which can be mapped to $\mathcal{A}$ by a homomorphism ($\mathcal{B} \preceq \mathcal{A}$) and those that cannot.

[4] We write $x = y$ instead of $=(x, y)$, $x = a$ instead of $=_a x$, and so on.

[5] Disjunctive or conjunctive normal form.

Definition 17. Let L be a logic. A σ-structure $\mathcal{A}$ has the $\mathsf{EHD}(L)$-*property*, if there is a computable function that maps every finite subsignature $\tau \subseteq \sigma$ to an L-sentence φ_τ s.t. for every countable τ-structure $\mathcal{B}$ holds: $\mathcal{B} \preceq \mathcal{A} \Leftrightarrow \mathcal{B} \models \varphi_\tau$.

We will see that for a negation-closed domain $\mathcal{D}$ with the $\mathsf{EHD}(\mathsf{BMWB})$-property, satisfiability of $\mathcal{ALC}^P(\mathcal{D})$ is decidable. For this reason, we are mainly interested in structures with $\mathsf{EHD}(L)$-property, where L is BMWB or some fragment of this logic. In this case, we might omit L and simply write EHD.

Remark 18. In [6, 7, 5] several (classes of) relational structures are investigated. Some of them that enjoy the property EHD are:

- the integers with equality-, order-, constants-, and modulo-constraints: $(\mathbb{Z}, =, <, (=_a)_{a\in\mathbb{Z}}, (\equiv_{a,b})_{a<b})$,
- the natural numbers with the same relational signature: $(\mathbb{N}, =, <, (=_a)_{a\in\mathbb{N}}, (\equiv_{a,b})_{0\leq a<b})$,
- the class of all semi-linear orders (see [5]),
- the class of all trees of height h for some fixed $h \in \mathbb{N}$,
- $(\mathbb{Z}^n, <_{\text{lex}}, =)$ where $<_{\text{lex}}$ is the lexicographic order,
- $\mathsf{Allen}_\mathbb{Z}$: the set of intervals over the integers together with Allen's relations, which allow to describe their relative positioning.

It was shown in [6] that $(\mathbb{Z}, <)$ has the EHD-property. Consider any countable $\{<\}$-structure $\mathcal{A} = (A, <)$. For $x, y \in A$ we write $x <^* y$ if there exist $x_1, \dots, x_n$ s.t. $x < x_1 < \cdots < x_n < y$ in $\mathcal{A}$ and call $\{x, x_1, \dots, x_n, y\}$ a $<$-*path* from x to y. It is proved that $\mathcal{A} \preceq (\mathbb{Z}, <)$ iff

- $\mathcal{A}$ is *acyclic*: there are no two elements x, y s.t. $x <^* y < x$, and
- for every two elements $x, y \in A$, there exists a bound n such that all $<$-path from x to y have at most n elements.

This can be expressed by the following BMWB-formulas: $\neg\exists x, y(\mathsf{reach}_<(x, y) \wedge y < x)$ and $\forall x \forall y\, \mathsf{B} X \mathsf{Path}(X, x, y)$. Here $\mathsf{reach}_<$ is the same as in Example 2, but with the edge relation E replaced by $<$, and $\mathsf{Path}(X, x, y)$ is a formula indicating that the set X is a $<$-path from x to y (see [7, Ex. 2]). Here the bounding quantifier limits the length of all paths between any two elements.

In [12, 13] concrete domains over the rationals are considered for the logics $\mathbb{Q}$-$\mathcal{SHIQ}$ and $\mathcal{TDL}$. The difference between $\mathcal{TDL}$ and $\mathcal{ALC}^P(\mathcal{D})$ is that the $\mathcal{TDL}$ only allows feature-paths as connectors to the concrete domain. For $\mathbb{Q}$-$\mathcal{SHIQ}$ and $\mathcal{TDL}$, it is stated that adding a unary predicate int, expressing that a certain concrete value has to be an integer, would be extremely useful. Decidability of reasoning in these logics under this addition remained an open problem. We show that the domain $\mathcal{Q} = (\mathbb{Q}, <, \mathsf{int}, \overline{\mathsf{int}})$, where $\mathsf{int} = \mathbb{Z}$ and $\overline{\mathsf{int}} = \mathbb{Q} \setminus \mathbb{Z}$, has the EHD-property. In [7, Lem. 38] it is shown that, if a domain $\mathcal{D}$ has the EHD-property, then so does $\mathcal{D}^=$, obtained by adding equality. This proves that $\mathcal{Q}^= = (\mathbb{Q}, =, <, \mathsf{int}, \overline{\mathsf{int}})$ (which is also negation-closed) has the EHD-property.

Proposition 19. *$\mathcal{Q} = (\mathbb{Q}, <, \mathsf{int}, \overline{\mathsf{int}})$ has the EHD-property.*

Sketch of proof. Let $\mathcal{A} = (A, <, \mathsf{int}^\mathcal{A}, \overline{\mathsf{int}}^\mathcal{A})$ be an arbitrary countable structure. We prove that $\mathcal{A}$ allows a homomorphism to $\mathcal{Q}$ iff

H1 $\mathcal{A}$ is acyclic,

H2 there exists no x s.t. $x \in \mathsf{int}^\mathcal{A} \cap \overline{\mathsf{int}}^\mathcal{A}$,

H3 given any two elements $x, y \in A$, there exists a bound n s.t. each $<$-path from x to y contains at most n elements from $\mathsf{int}^\mathcal{A}$.

In this setting, it is only the number of elements of $\mathsf{int}^\mathcal{A}$ that needs to be bounded on all paths between two elements. The reason is that, being $\mathbb{Q}$ dense, we can accommodate any countable amount of numbers in any interval, provided that they are not forced to be integers. Properties H1-H3 are easily defined in BMWB: acyclicity is expressed as above using $\mathsf{reach}_<$, H2 is given by $\neg\exists x(\mathsf{int}(x) \wedge \overline{\mathsf{int}}(x))$ and H3 by $\forall x,y\, \mathsf{B} X[X \subseteq \mathsf{int}^\mathcal{A} \wedge \exists Z(X \subseteq Z \wedge \mathsf{Path}(Z, x, y))]$. □

4.2 Satisfiability of $\mathcal{ALC}^P(\mathcal{D})$

We are now ready to state our main result:

Theorem 20. *If a concrete domain $\mathcal{D}$ is negation-closed and has the property $\mathsf{EHD}(\mathsf{BMWB})$, the satisfiability problem for $\mathcal{ALC}^P(\mathcal{D})$ is decidable.*

This theorem classifies all the concrete domains listed in Remark 18 and the new one from Proposition 19 positively, yielding a good number of decidability results for $\mathcal{ALC}^P(\mathcal{D})$ w.r.t. general TBoxes, which strictly improves what was known so far.

The idea behind the proof of this theorem is to separate the search of a $\mathcal{D}$-interpretation for a concept C w.r.t. a TBox $\mathcal{T}$ into two parts: In a first step look for an ordinary $\mathcal{ALC}$ interpretation (i.e., without the valuation function) that is a model for an *abstracted* version of C and $\mathcal{T}$. That is, replace each atomic constraint appearing in C and $\mathcal{T}$ with a fresh concept name B and obtain a classical $\mathcal{ALC}$-concept C_a and TBox $\mathcal{T}_a$, where the a stands for 'abstracted'. The fact that C_a is satisfiable w.r.t. $\mathcal{T}_a$ is clearly not enough to guarantee that C is satisfiable w.r.t. $\mathcal{T}$. For instance, the $\mathcal{ALC}^P(\mathcal{D})$-concept $\exists r.(x < Sx \;\wedge\; Sx < x)$ is unsatisfiable, while its abstraction $\exists r.(B_1 \sqcap B_2)$ is not. To avoid this effect, the second step creates from the model of the abstracted concept a so-called *constraint graph*, a structure for storing the information from the constraints that were abstracted away. It turns out that if such constraint graph allows a homomorphism to our concrete domain, then this guarantees that the constraints are satisfied.

For the rest of this section let us fix a signature σ, a negation-closed σ-structure $\mathcal{D}$ as concrete domain with the EHD-property, and an $\mathcal{ALC}^P(\mathcal{D})$-concept C and TBox $\mathcal{T}$, both in constraint normal form, in which only the atomic constraints $\theta_1, \dots, \theta_n$ occur. Let d_i be the depth of each θ_i, and let $B_1, \dots, B_n \in \mathsf{N_C} \setminus \mathsf{Sub}(\mathcal{T}, C)$.

Definition 21. Let $P = r_1 \dots r_p$ and c be a conjunction of the atomic constraints $\theta_1, \dots, \theta_m$ with $m \leq n$ with depths s.t. $0 =: d_0 \leq d_1 \leq \cdots \leq d_m \leq d_{m+1} := p$ (if this is not the case, it suffices to reorder the constraints). Define the *abstraction of an existential path constraint* $E = \exists P.c(S^{i_1}x_1, \dots, S^{i_k}x_k)$, as

$$E_a = \exists P_1.(B_1 \sqcap \exists P_2.(B_2 \sqcap \cdots \exists P_m.(B_m \sqcap \exists P_{m+1}.\top)\dots)\,, \quad (1)$$

where $\exists P_i$ is short for $\exists r_{d_{i-1}+1} \dots \exists r_{d_i}$. If $d_i = d_{i+1}$, then $\exists P_{i+1}$ is empty. Let c' be a disjunction of atomic constraints containing $\theta_1, \dots, \theta_m$ with $0 =: d_0 \leq d_1 \leq \cdots \leq d_m \leq d_{m+1} := p$. Define the *abstraction of a universal path constraint* $E = \forall P.c'(S^{i_1}x_1, \dots, S^{i_k}x_k)$ as

$$E_a = \forall P_1.(B_1 \sqcup \forall P_2.(B_2 \sqcup \cdots \forall P_m.(B_m \sqcup \forall P_{m+1}\bot)\dots)\,. \quad (2)$$

We define C_a and $\mathcal{T}_a$ as the $\mathcal{ALC}$-concept and TBox obtained by C and $\mathcal{T}$ by replacing every occurrence of a path constraint E by its abstraction E_a.

Let us consider $C = \exists.r_1r_2r_3(x = y \wedge x < S^2x \wedge S^1y = S^2x)$, its abstraction C_a is $B_1 \sqcap \exists r_1.\exists r_2.(B_2 \sqcap B_3 \sqcap \exists r_3.\top)$, where the new concept names are assigned to the atomic constraints in order of appearance. Notice how we use the locality of constraints from $\mathcal{ALC}^P(\mathcal{D})$ to individuate the 'lower' node involved in the constraint (the one at depth d_i) and mark it as belonging to the fresh concept B_i. This way, when considering a tree-model of the abstracted concept C_a w.r.t. $\mathcal{T}_a$, all paths of length d_i that end in a node marked with B_i should satisfy the constraint θ_i.

Definition 22. Given an n-tree $\mathcal{D}$-interpretation $\mathcal{I} = ([1,n]^*, \cdot^{\mathcal{I}}, \gamma_{\mathcal{I}})$ s.t. $B_1{}^{\mathcal{I}} = \dots = B_m{}^{\mathcal{I}} = \emptyset$, we define the *abstraction of* $\mathcal{I}$ as the interpretation $\mathcal{I}_a = ([1,n]^*, \cdot^{\mathcal{I}_a})$, where $\cdot^{\mathcal{I}_a}$ is defined as

- $A^{\mathcal{I}_a} = A^{\mathcal{I}}$ for all $A \in (\mathsf{N_C} \setminus \{B_1, \dots, B_m\})$,
- $r^{\mathcal{I}_a} = r^{\mathcal{I}}$ for all $r \in \mathsf{N_R}$,
- if $\theta_j = R(S^{i_1}x_1, \dots, S^{i_k}x_k)$ has depth d_j, then $u \in B_j{}^{\mathcal{I}_a}$ iff
  - $u = wv$ for some $w, v \in [1,n]^*$ with $|v| = d_j$, and
  - $(\gamma(wv_1, x_1), \dots, \gamma(wv_k, x_k)) \in R^{\mathcal{D}}$,

  where v_t denotes the prefix of v of length i_t.

Hence, the fact that an element wv with $|v| = d_j$ belongs to the interpretation of B_j means that the atomic constraint θ_j is satisfied along every path that starts in node w and descends in the tree via wv. Now let $\mathsf{Reg}_{C,\mathcal{T}}$ be the set of register variables occurring in C and $\mathcal{T}$.

Definition 23. Take an n-tree interpretation $\mathcal{J} = ([1,n]^*, \cdot^{\mathcal{J}})$ where $B_1^{\mathcal{J}}, \dots, B_m^{\mathcal{J}}$ can be non-empty. We define the *constraint graph* $\mathcal{G}_{\mathcal{J}}$ of $\mathcal{J}$ as a countable σ-structure $\mathcal{G}_{\mathcal{J}} = (([1,n]^* \times \mathsf{Reg}_{C,\mathcal{T}}), R_1^{\mathcal{G}}, R_2^{\mathcal{G}}, \dots)$ as follows: The interpretation $R^{\mathcal{G}}$ of the relation $R \in \sigma$ contains all k-tuples $((wv_1, x_1), \dots, (wv_k, x_k))$, where $k = \mathsf{ar}(R)$, for which there are $1 \le j \le m$ and $v \in [1,n]^{d_j}$ s.t. $wv \in B_j^{\mathcal{J}}$, and $\theta_j = R(S^{i_1}x_1, \dots, S^{i_k}x_k)$, where v_t still denotes the prefix of v of length i_t.

The domain of $\mathcal{G}_{\mathcal{J}}$ has one element for each pair (v, x) where v is a member of the domain of $\mathcal{J}$ and x is a register variable appearing in C. When we abstract an atomic constraint θ_i we replace it with its *placeholder* B_i, but any occurrence of B_i marks a path where θ_i needs to hold. Such information is stored in the relations of $R^{\mathcal{G}}$.

Example 24. Let $\mathcal{D}$ be a concrete domain having $<$ and $=$ in its signature. Suppose the concept names B_1 and B_2 are used to replace the atomic constraints $\theta_1 = (x = y)$ and $\theta_2 = (x < Sx)$ of depth $d_1 = 0$ and $d_2 = 1$, respectively. Figure 1 depicts the constraint graph associated with an ordinary 2-tree interpretation $\mathcal{J}$.

In the next theorem, we illustrate the connection between the satisfiability of an $\mathcal{ALC}^P(\mathcal{D})$-concept w.r.t. a TBox, and the satisfiability of its abstraction. We denote by $\#_E(\mathcal{T}, C)$ the number of existentially quantified subconcepts that occur in $\mathsf{Sub}(\mathcal{T}, C)$. Let C be an $\mathcal{ALC}^P(\mathcal{D})$-concept and $\mathcal{T}$ a TBox—both in constraint normal form—and let $n = d \cdot \#_E(C, \mathcal{T})$ where d is the maximum depth of all constraints appearing in $\mathsf{Sub}(\mathcal{T}, C)$. Then the following holds:

Theorem 25. *C is satisfiable w.r.t. $\mathcal{T}$ iff there exists an ordinary n-tree interpretation $\mathcal{I} = ([1,n]^*, \cdot^{\mathcal{I}})$ s.t. $\mathcal{I} \models_{\mathcal{T}_a} C_a$ and s.t. $\mathcal{G}_{\mathcal{I}} \preceq \mathcal{D}$.*

Proof. Let $\theta_1, \dots, \theta_m, d_1, \dots, d_m$ and $\mathsf{Reg}_{C,\mathcal{T}}$ be as before.

($\Rightarrow$) W.l.o.g. assume that $\mathcal{I} = ([1,n]^*, \cdot^{\mathcal{I}}, \gamma_{\mathcal{I}})$ is an n-tree $\mathcal{D}$-interpretation s.t. $\mathcal{I} \models_{\mathcal{T}} C$. Our first claim is that $\mathcal{I}_a \models_{\mathcal{T}_a} C_a$, which we show by induction on the structure of C, proving that for all $v \in [1,n]^*$ and for all subconcepts $E \in \mathsf{Sub}(\mathcal{T}, C)$, $u \in E^{\mathcal{I}}$ implies $u \in E_a^{\mathcal{I}_a}$. Due to space limitations, we show only some cases. The remaining ones are shown in [8].

- If $E \in \mathsf{Sub}(\mathcal{T}, C)$ is a concept name, then $E_a = E$.
- If $E = F \sqcap G$, then $u \in E^{\mathcal{I}}$ implies $u \in F^{\mathcal{I}}$ and $u \in G^{\mathcal{I}}$. By induction hypothesis we have that $u \in F_a^{\mathcal{I}_a}$ and $u \in G_a^{\mathcal{I}_a}$ which yields $u \in (F_a \sqcap G_a)^{\mathcal{I}_a} = E_a^{\mathcal{I}_a}$.
- If $E = \exists r.F$ and $u \in E^{\mathcal{I}}$, then there exists an element $v \in [1,n]^*$, s.t. $(u,v) \in r^{\mathcal{I}}$ and $v \in F^{\mathcal{I}}$. Then $(u,v) \in r^{\mathcal{I}_a}$ by definition of $\mathcal{I}_a$ and $v \in F_a^{\mathcal{I}_a}$ by induction hypothesis. Together we obtain $u \in (\exists r.F_a)^{\mathcal{I}_a} = E_a^{\mathcal{I}_a}$.
- Let $E = \exists P.c(S^{i_1}x_1, \dots, S^{i_t}x_t)$ with $P = r_1 \cdots r_p$. Since C and $\mathcal{T}$ are in constraint normal form, we can assume that (eventually renaming the atomic constraints) $c = \theta_1 \wedge \dots \wedge \theta_n$ where the depths $d_1, \dots, d_n$ satisfy that $0 =: d_0 \le d_1 \le \dots \le d_n \le d_{n+1} := p$. Since $u \in E^{\mathcal{I}}$, we know that there exists a tuple $(u_0, \dots, u_p) \in P^{\mathcal{I}}$ s.t. $u_0 = u$ and that $\mathcal{D} \models c\big(\gamma(u_{i_1}, x_1), \dots, \gamma(u_{i_t}, x_t)\big)$. If we have $\theta_i = R(S^{j_1}y_1, \dots, S^{j_k}y_k)$, this means that $(\gamma(u_{j_1}, y_1), \dots, \gamma(u_{j_k}, y_k)) \in R^{\mathcal{D}}$. By definition of $\mathcal{I}_a$, this implies that $u_{d_i} \in B_i^{\mathcal{I}_a}$. Now, since $(a,b) \in r^{\mathcal{I}}$ implies $(a,b) \in r^{\mathcal{I}_a}$, then $(u_0, \dots, u_p) \in P^{\mathcal{I}_a}$ as well. This, together with the fact that $u_{d_i} \in B_i^{\mathcal{I}_a}$ for $i = 1, \dots, n$, implies that $u \in (\exists P_1.(B_1 \sqcap \exists P_2.(B_2 \sqcap \dots \exists P_n.(B_n \sqcap \exists P_{n+1}.\top) \dots))^{\mathcal{I}_a}$, where $\exists P_i$ is short for $\exists r_{d_{i-1}+1} \dots \exists r_{d_i}$. This shows the claim.

The second claim is that $\mathcal{G}_{\mathcal{I}_a} \preceq \mathcal{D}$. Specifically, we want to prove that the valuation function

$$\gamma_{\mathcal{I}} : ([1,n]^* \times \mathsf{Reg}_{C,\mathcal{T}}) \to D$$

is a homomorphism. For this, suppose that there is a tuple $\big((u_1, x_1), \dots, (u_k, x_k)\big) \in R^{\mathcal{G}}$. By Definition 23 this means that there exist $j \in \{1, \dots, m\}$ and $wv \in (B_j)^{\mathcal{I}_a}$ s.t. θ_j has the form $R(S^{i_1}x_1, \dots, S^{i_k}x_k)$ with depth d_j and s.t. $v = v_1 \cdots v_{d_j}$ and $u_t = wv_{i_t}$ for all $t = 1 \dots k$. By Definition 22, this means that $(\gamma_{\mathcal{I}}(u_1, x_1), \dots, \gamma_{\mathcal{I}}(u_t, x_t)) \in R^{\mathcal{D}}$, as wanted.

($\Leftarrow$) Now we show that, given an ordinary n-tree interpretation $\mathcal{I} = ([1,n]^*, \cdot^{\mathcal{I}})$ s.t. $\mathcal{I} \models_{\mathcal{T}_a} C_a$ and a homomorphism h from $\mathcal{G}_{\mathcal{I}}$ to $\mathcal{D}$, we can construct a $\mathcal{D}$-interpretation $\mathcal{J}$ s.t. $\mathcal{J} \models_{\mathcal{T}} C$. Let's define $\mathcal{J} = ([1,n]^*, \cdot^{\mathcal{J}}, h)$, where $\cdot^{\mathcal{J}}$ coincides with $\cdot^{\mathcal{I}}$ on all concept and role names, and is extended to all concepts using the valuation function h. We can again prove by induction that, for all concepts $E \in \mathsf{Sub}(\mathcal{T}, C)$ and for all $u \in [1,n]^*$, $u \in (E_a)^{\mathcal{I}}$ implies $u \in E^{\mathcal{J}}$ (we show only one interesting case):

Suppose $E = \exists P.c(S^{i_1}x_1, \dots, S^{i_k}x_k)$ where $P = r_1 \cdots r_p$ is a role-path of length p and c is a conjunction of atomic constraints $\theta_1 \wedge \dots \wedge \theta_n$ with depths $d_1, \dots, d_n$ such that $0 =: d_0 \le d_1 \le \dots \le d_n \le d_{n+1} := p$. Then the abstraction of E is

$$E_a = \exists P_1.(B_1 \sqcap \exists P_2.(B_2 \sqcap \dots \exists P_n.(B_n \sqcap \exists P_{n+1}.\top) \dots)$$

with $P_i = r_{d_{i-1}+1} \cdots r_{d_i}$. If $u \in (E_a)^{\mathcal{I}}$, then there exists a tuple $(u_0, \dots, u_p) \in P^{\mathcal{I}}$ with $u_0 = u$ and s.t. $u_{d_i} \in (B_i)^{\mathcal{I}}$ for $i = 1, \dots, n$. Fix $i \in \{1, \dots, n\}$, if θ_i has the form $R(S^{j_1}y_1, \dots, S^{j_t}y_t)$, according to Definition 23, this means that $((u_{j_1}, y_1), \dots, (u_{j_t}, y_t)) \in R^{\mathcal{G}}$. Now, since h is a homomorphism from $\mathcal{G}_{\mathcal{I}}$ to $\mathcal{D}$, we have $(h(u_{j_1}, y_1), \dots, h(u_{j_t}, y_t)) \in R^{\mathcal{D}}$, which means that $\mathcal{D} \models R(h(u_{j_1}, y_1), \dots, h(u_{j_t}, y_t))$. Since this is true for an arbitrary $i \in \{1, \dots, n\}$ this holds true for the conjunction $\theta_1 \wedge \dots \wedge \theta_n$, that is $\mathcal{D} \models c(h(u_{i_1}, x_1), \dots, h(u_{i_k}, x_k))$. Also, since $r^{\mathcal{I}} = r^{\mathcal{J}}$, we know that $(u_0, \dots, u_p) \in P^{\mathcal{J}}$.
This means that $u \in E^{\mathcal{J}}$, as wanted. □

We are now almost ready to give the proof of the main result, of this paper: Theorem 20. We only need a few additional results.

Definition 26. Given an n-tree ordinary interpretation $\mathcal{I} = ([1,n]^*, \cdot^{\mathcal{I}})$, we define an *$n$-tree $T(\mathcal{I})$ over the signature* $\{S\} \cup \mathsf{N_C} \cup \mathsf{N_R}$, where $\mathsf{N_C}$ and $\mathsf{N_R}$ are seen as unary predicates whose interpretation is given by: $A^{T(\mathcal{I})} = A^{\mathcal{I}}$ for each $A \in \mathsf{N_C}$ and $r^{T(\mathcal{I})} = \{xi \in [1,n]^* \mid (x, xi) \in r^{\mathcal{I}}\}$ for all $r \in \mathsf{N_R}$.

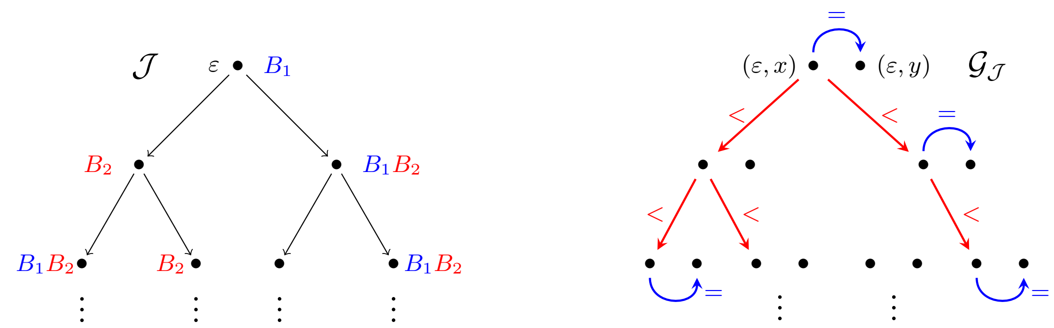

Figure 1. An ordinary 2-tree interpretation (where B_1 and B_2 have non-empty interpretations) and its associated constraint graph $\mathcal{G}_\mathcal{J}$ from Example 24.

Remark 27. The only difference between an n-tree interpretation $\mathcal{I}$ and its induced n-tree $T = T(\mathcal{I})$ is that roles are turned into unary predicates s.t., if a pair $(x, y) \in r^\mathcal{I}$, now $y \in r^T$. In particular, if we define $\mathcal{G}_T = (([1, n]^* \times \mathsf{Reg}_{C,\mathcal{T}}), R_1^\mathcal{G}, R_2^\mathcal{G}, \dots)$ (the *constraint graph* of T) exactly as in Definition 23, only substituting the interpretation $\mathcal{J}$ with T, then $\mathcal{G}_\mathcal{I} = \mathcal{G}_T$.

Lemma 28. *Given C and $\mathcal{T}$ an $\mathcal{ALC}$-concept and TBox in nnf, we can write a* FO*-formula φ over the signature $\{S\} \cup \mathsf{N_C} \cup \mathsf{N_R}$, where all elements of $\mathsf{N_C} \cup \mathsf{N_R}$ are seen as unary symbols, s.t. for any n-tree interpretation $\mathcal{I} = ([0, 1]^*, \cdot^\mathcal{I})$, $\mathcal{I} \models_\mathcal{T} C$ if and only if $T(\mathcal{I}) \models \varphi$.*

Proof. The method is similar to the one in [2, Chapter 3], with the only difference that here roles and features are seen as unary predicates added to the second node of the relation. This can be safely done due to the use of tree-shaped models. We define two translations π_x and π_y which inductively map $\mathcal{ALC}$-concepts to FO formulas with only one free variable, x or y respectively:

- $\pi_i(A) := A(i)$ for each $A \in \mathsf{N_C}$ and $i = x, y$;
- $\pi_i(\neg A) := \neg A(i)$ for each $A \in \mathsf{N_C}$ and $i = x, y$;
- $\pi_i(D \sqcap E) := \pi_i(D) \wedge \pi_i(E)$ for $i = x, y$;
- $\pi_i(D \sqcup E) := \pi_i(D) \vee \pi_i(E)$ for $i = x, y$;
- $\pi_i(\exists r.D) := \exists j.S(i,j) \wedge r(j) \wedge \pi_j(D)$ for $(i,j) = (x,y)$ or (y,x);
- $\pi_i(\forall r.D) := \forall j.(S(i,j) \wedge r(j)) \rightarrow \pi_j(D)$ for $(i,j) = (x,y)$ or (y,x);

Now let $R_\mathcal{T}$ be the set of role names appearing in C and $\mathcal{T}$, and let $F \subseteq R_\mathcal{T}$ be feature names. Keep in mind that the root ε of a tree is definable in FO. We define

$$\psi_{R_\mathcal{T}} := \forall (x \neq \varepsilon). \bigvee_{r \in R_\mathcal{T}} r(x) \ \wedge \bigwedge_{r,s \in R_\mathcal{T}, r \neq s} \neg (r(x) \wedge s(x))$$

$$\psi_F := \forall x. \forall (y \neq z).\ S(x,y) \wedge S(x,z) \rightarrow \bigwedge_{f \in F} \neg (f(y) \wedge f(z))\,.$$

The formula $\psi_{R_\mathcal{T}}$ enforces that each pair of elements (x, y), where y is a successor of x, is assigned a unique role name. ψ_F ensures that the functionality of the features is respected. Then we can prove easily that given a tree-shaped interpretation $\mathcal{I}$ and a TBox $\mathcal{T} = \{C_\mathcal{T}\}$, $\mathcal{I} \models_\mathcal{T} C$ if and only if $T(\mathcal{I})$ is a model for the following FO formula $\varphi = \exists x.\pi_x(C) \wedge \forall x.\pi_x(C_\mathcal{T}) \wedge \psi_{R_\mathcal{T}} \wedge \psi_F$. □

Lemma 29. *Let $C, \mathcal{T}$ and φ be as in Lemma 28. Given an n-tree T over the relational signature $\{S\} \cup \mathsf{N_R} \cup \mathsf{N_F}$ that satisfies φ we can build an n-tree interpretation $\mathcal{I}$ s.t. $T = T(\mathcal{I})$ and s.t. $\mathcal{I} \models_\mathcal{T} C$.*

Sketch of proof. The fact that $T \models \varphi$ means in particular that $T \models \psi_{R_\mathcal{T}} \wedge \psi_F$, which guarantees that each node of the tree T is assigned at most one role name, and that the functionality of the features is respected. We can therefore safely define $A^\mathcal{I} = A^T$ for all $A \in \mathsf{N_C}$ and $r^\mathcal{I} = \{(x, y) \in ([1, n]^*)^2 \mid S(x, y) \text{ and } y \in r^T\}$ for all $r \in \mathsf{N_R}$ and obtain a tree shaped interpretation. It is easy to see that $T(\mathcal{I}) = T$, and $\mathcal{I} \models_\mathcal{T} C$ can be proved by structural induction. □

Next we show a useful property of BMWB, which is also needed to prove our main result.

Definition 30. Let $k \in \mathbb{N}$ and let $\mathcal{A} = (A, R_1^\mathcal{A}, R_2^\mathcal{A}, \dots)$ be a structure over the signature σ that does not contain relation symbols $\sim, P_1, P_2, \dots, P_k$ ($\sim$ is binary and all P_i are unary). The *k-copy of* $\mathcal{A}$, denoted by $\mathcal{A}^{\times k}$, is the $(\sigma \cup \{\sim, P_1, P_2, \dots, P_k\})$-structure with domain $(A \times \{1, 2, \dots, k\})$ and

- for all $R \in \sigma$ if R has arity m, $R^{\mathcal{A}^{\times k}}$ is defined as
$$\{((a_1, i), \dots, (a_m, i)) \mid (a_1, \dots, a_m) \in R^\mathcal{A}, 1 \leq i \leq k\}\ ,$$
- $\sim^{\mathcal{A}^{\times k}} = \{((a, i_1), (a, i_2)) \mid a \in A, 1 \leq i_1, i_2 \leq k\}$, and
- for each $1 \leq m \leq k$, ${P_m}^{\mathcal{A}^{\times k}} = \{(a, m) \mid a \in A\}$.

Given a structure $\mathcal{A}$, the k-copy operation creates a new structure, $\mathcal{A}^{\times k}$, which contains k many copies of $\mathcal{A}$: there are k disjoint substructures of $\mathcal{A}^{\times k}$ (identifiable through the predicates $P_1, \dots, P_k$) which, seen as σ-structures, are isomorphic to $\mathcal{A}$. The additional binary predicate $\sim$ relates all those members of $\mathcal{A}^{\times k}$ which are a duplicate of the same element in $\mathcal{A}$.

The following proposition states that BMWB is *compatible* with the k-copy operation, i.e., whatever property is specified on $\mathcal{A}^{\times k}$ using BMWB can also be recognized by BMWB directly on $\mathcal{A}$.

Proposition 31 (Prop. 2.26 of [4]). *Let $k \in \mathbb{N}$, $\mathcal{A}$ some infinite structure over the signature σ, and $\tau = \sigma \cup \{\sim, P_1, P_2, \dots, P_k\}$ where $\sim$ is a fresh binary relation symbol and k fresh unary relation symbols $P_1, \dots, P_k$. Given a* BMWB*-sentence φ over τ, we can compute a* BMWB*-sentence φ^k over σ s.t. $\mathcal{A}^{\times k} \models \varphi$ iff $\mathcal{A} \models \varphi^k$.*

Let $\tau \subseteq \sigma$, we say a τ-structure $\mathcal{A}$ with domain A is FO-*interpretable* in a σ-structure $\mathcal{B}$ with domain B, if there exists a FO-formula φ such that $A \cong \{b \in B \mid \mathcal{B} \models \varphi(b)\}$, and for each $R \in \tau$ of arity k, there exists an FO-formula φ_R such that $R^\mathcal{A} \cong \{(b_1, \dots, b_k) \in B^k \mid \mathcal{B} \models \varphi_R(b_1, \dots, b_k)\}$. Intuitively, we can use FO to *describe* in $\mathcal{B}$ a substructure that is isomorphic to $\mathcal{A}$.

Lemma 32. *Suppose $\mathsf{Reg}_{C,\mathcal{T}} = \{x_1, \dots, x_k\}$, then for an n-tree T over the signature $\{S\} \cup \mathsf{N_C} \cup \mathsf{N_R}$, $\mathcal{G}_T$ is* FO*-interpretable in $T^{\times k}$.*

Proof. The domains of $\mathcal{G}_T$ and $T^{\times k}$, $([1, n]^* \times \{x_1, \dots, x_k\})$ and $([1, n]^* \times \{1, 2, \dots, k\})$ respectively, are in a bijection via the mapping $f : (v, x_k) \mapsto (v, k)$. We extend the bijection f to tuples of elements of $([1, n]^* \times \{x_1, \dots, x_k\})$ as $f(a_1, \dots, a_t) = (f(a_1), \dots, f(a_t))$.

We claim that the relations $R_1^{\mathcal{G}}, R_2^{\mathcal{G}}, \dots$ from $\mathcal{G}_T$ can be represented in $T^{\times k}$ using FO. We describe how: suppose the relation $R \in \sigma$ of arity t is used to form one of the atomic constraints $\theta = R(S^{i_1}y_1, \dots, S^{i_t}y_t)$, with $y_1, \dots, y_t \in \{x_1, \dots, x_k\}$ and $d = \max\{i_1, \dots, i_t\}$. Then we know that a tuple $\big((v_1, y_1), \dots, (v_t, y_t)\big)$ belongs to $R^{\mathcal{G}}$ if (1) there exist elements $w_0, w_1, \dots, w_d \in [1, n]^*$ s.t. $w_{i_l} = v_l$ for $l = 1, \dots, t$ and $S(w_{j-1}, w_j)$ holds in T for $j = 1, \dots, d$, and (2) $v_d \in B_j^T$. We would like to identify the tuples in $T^{\times k}$ in bijection through f with those tuples in $\mathcal{G}_T$ satisfying Conditions (1) and (2). These are the ones that satisfy the following FO-formula

$$\varphi_\theta(a_1, \dots, a_t) = \exists b_0 \dots \exists b_d \bigwedge_{j=1,\dots,d} S(b_{j-1}, b_j) \wedge \bigwedge_{l=1,\dots,t} b_{i_l} \sim a_l \wedge \bigwedge_{i=1,\dots,t} P_{z_i}(a_i)$$

where $i_1, \dots, i_t$ are the same indices appearing in $\theta = R(S^{i_1}y_1, \dots, S^{i_t}y_t)$ and z_i is s.t. $y_i = x_{z_i}$. Once φ_{θ_j} is defined for the atomic constraints $\theta_1, \dots, \theta_n$, which appear in C and $\mathcal{T}$, we state the following: if the relation R of arity t is used in all and only $\theta_{j_1}, \dots, \theta_{j_k}$, then $\bar{a} = (a_1, \dots, a_t) \in R^{\mathcal{G}}$ if and only if $\varphi_R(f(\bar{a})) = \varphi_{\theta_{j_1}}(f(\bar{a})) \vee \dots \vee \varphi_{\theta_{j_k}}(f(\bar{a}))$ holds. If the relation $R \in \sigma$ of arity t is not used in any of the atomic constraints $\theta_{j_1}, \dots, \theta_{j_k}$, then there will be no tuple in $\mathcal{G}_T$ which belongs to $R^{\mathcal{G}}$. Therefore $(a_1, \dots, a_t) \in R^{\mathcal{G}}$ iff $\varphi_R(f(\bar{a})) = \bot$ holds. □

Corollary 33 (of Lemma 32)**.** *If α is a* BMWB*-formula over the signature σ, we can write a* BMWB *formula α' over the signature $\{S\} \cup \mathsf{N_C} \cup \mathsf{N_R} \cup \{\sim, P_1, P_n\}$ s.t. $\mathcal{G}_T \models \alpha$ if and only if $T^{\times k} \models \alpha'$.*

Sketch of proof. The formula α' is obtained from α by replacing any occurrence of a formula $R(a_1, \dots, a_t)$ by the formula $\varphi_R(a_1, \dots, a_t)$ defined in the proof of Lemma 32. □

We are now finally ready to give the proof of our main result.

Proof of Thm. 20. Let C be an $\mathcal{ALC}^P(\mathcal{D})$-concept and $\mathcal{T}$ a TBox respectively. Let $n = d \cdot \#_E(\mathcal{T}, C)$ where d is the maximum depth of all constraints that appear in $\mathsf{Sub}(\mathcal{T}, C)$. Due to Lemma 15 we can assume w.l.o.g., that C and $\mathcal{T}$ are in constraint normal form. By Theorem. 25, we have to check, whether there is an ordinary n-tree interpretation $\mathcal{I}$ s.t. $\mathcal{I} \models_{\mathcal{T}_a} C_a$ and $\mathcal{G}_\mathcal{I} \preceq \mathcal{D}$.

Let $\tau \subseteq \sigma$ be the finite subsignature consisting of all relation symbols that occur in C and $\mathcal{T}$. Note that $\mathcal{G}_\mathcal{I}$ is actually a countable τ-structure. Since the concrete domain $\mathcal{D}$ has the property EHD(BMWB), one can compute from τ a BMWB-sentence α s.t. for every countable τ-structure $\mathcal{B}$ we have $\mathcal{B} \models \alpha$ iff $\mathcal{B} \preceq \mathcal{D}$. Our new goal is to decide whether there is an ordinary n-tree interpretation $\mathcal{I}$ s.t.

$$\mathcal{I} \models_{\mathcal{T}_a} C_a \text{ and } \mathcal{G}_\mathcal{I} \models \alpha\,. \tag{3}$$

Now $\mathcal{T}_a$ and C_a are ordinary $\mathcal{ALC}$-concepts. We can use Lemma 28, Lemma 29 and Remark 27, and obtain a FO formula φ s.t. if C_a is satisfied w.r.t. $\mathcal{T}_a$ by some n-tree interpretation $\mathcal{I}$, then φ is satisfied by an n-tree T s.t. $\mathcal{G}_\mathcal{I} = \mathcal{G}_T$. Also, if φ is satisfied by some n-tree T, then there exists an n-tree interpretation $\mathcal{I}$ s.t. $\mathcal{I} \models_{\mathcal{T}_a} C_a$ and s.t. $\mathcal{G}_T = \mathcal{G}_\mathcal{I}$.

Then finding $\mathcal{I}$ s.t. (3) holds is equivalent to finding an n-tree T s.t. $T \models \varphi$ and $\mathcal{G}_T \models \alpha$. By Corollary 33, we can find a BMWB-formula β s.t. $\mathcal{G}_T \models \alpha$ iff $T^{\times k} \models \beta$. But we also know, due to Proposition 31, that we can compute a formula β^k s.t. $T^{\times k} \models \beta$ iff $T \models \beta^k$. At this point we have to check whether there exists an n-tree T s.t. $T \models \varphi \wedge \beta^k$, where $\psi \wedge \beta^k$ is a BMWB-sentence. By Theorem 3 this is decidable, which completes the proof. □

5 Conclusion and Future Work

We have introduced a novel way to integrate concrete domains in $\mathcal{ALC}$, via *path constraints*. The resulting logic, $\mathcal{ALC}^P(\mathcal{D})$, is of incomparable expressiveness with the several variants of $\mathcal{ALC}(\mathcal{D})$ that are present in the literature. We have seen, however, how on the domains that we are interested in, our logic is strictly more expressive: $\mathcal{ALC}^P(\mathcal{D})$ allows not only feature-paths, but also full role-paths, to connect abstract individuals and their concrete attributes.

We exploit the path-structure of the constraints to show that $\mathcal{ALC}^P(\mathcal{D})$ is compatible with the EHD-method from [6] and show the very general result: satisfiability for $\mathcal{ALC}^P(\mathcal{D})$ is decidable w.r.t. general TBoxes, if the concrete domain $\mathcal{D}$ is negation-closed and has the EHD-property. This solves the problem that has been open for more than a decade (see [13]), whether reasoning in $\mathcal{ALC}$ with non-dense concrete domains such as the natural numbers or the integers would be decidable in the presence of general TBoxes, since these domains enjoy our required properties. Such domains did not satisfy the ω-admissibility criterion that was formulated in [16]. In this sense, we prove that this ω-admissibility is not a necessary condition to guarantee the decidability of reasoning over a concrete domain in the presence of general TBoxes.

We could have easily chosen a more expressive DL than $\mathcal{ALC}$ as underlying logic. In principle we could add any concept constructor preserving the tree model property, and that can be then translated to MSO over trees with one successor and unary predicates only (see lemma 28). Examples of such constructors would be transitive closure, role hierarchy and qualified number restriction.

The main open question remains the complexity. The EHD method is a reduction to satisfiability of WMSO+B over infinite binary trees, which is shown to be decidable in [3]. Here the authors do not provide complexity bounds for their decision procedure. On the other hand, the WMSO+B-formulas that need to be checked for decidability are fixed and depend solely on the concrete domain. Roughly speaking, once we fix our domain $\mathcal{D}$, the EHD method transforms a given $\mathcal{ALC}^P(\mathcal{D})$-TBox and -concept into a *constraint normal form* which already results in a blow-up of the size. This in turn get transformed into an MSO-formula φ (which is clearly non optimal). We then have to decide whether a conjunction of φ and a fixed WMSO+B-formula ψ (which depends on $\mathcal{D}$) is decidable. Analyzing this procedure would very hardy lead to tight complexity bounds. In our opinion the EHD-method is more of an admissibility criterion, which provides easy conditions on a concrete domain $\mathcal{D}$ to establish whether reasoning with it remains decidable or not.

Also, it would be interesting to know if one can add constant predicates of the form $(=_q)_{q\in\mathbb{Q}}$ to the domain $\mathcal{Q}$ from Prop. 19 and prove that the resulting structure still has the EHD-property. We conjecture that a method similar to the one presented in [7] for constant predicates over the integers could be applied to this case.

Another follow-up question is whether the EHD method be can adapted to show decidability for *fuzzy* concrete domains, similarly as it was shown in [17] for the criterion of ω-admissibility.

ACKNOWLEDGEMENTS

This work is supported by the German Research Foundation (DFG) within the Collaborative Research Center SFB 912 – HAEC and the Graduiertenkolleg 1763 (QuantLA).

REFERENCES

[1] Franz Baader and Phillip Hanschke, 'A scheme for integrating concrete domains into concept languages', in *Proceedings of the 12th International Joint Conference on Artificial Intelligence, IJCAI-91*, pp. 452–457, Sydney (Australia), (1991).

[2] Franz Baader, Ian Horrocks, and Ulrike Sattler, 'Description logics', in *Handbook of Knowledge Representation*, eds., F. van Harmelen, V. Lifschitz, and B. Porter, 135–179, Elsevier, (2007).

[3] Mikołaj Bojańczyk and Szymon Toruńczyk, 'Weak MSO+U over infinite trees', in *29th International Symposium on Theoretical Aspects of Computer Science (STACS 2012)*, eds., Christoph Dürr and Thomas Wilke, volume 14 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 648–660, Dagstuhl, Germany, (2012). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[4] Claudia Carapelle, *On the satisfiability of temporal logics with concrete domains*, Ph.D. dissertation, Universität Leipzig, 2015. http://nbn-resolving.de/urn:nbn:de:bsz:15-qucosa-190987.

[5] Claudia Carapelle, Shiguang Feng, Alexander Kartzow, and Markus Lohrey, 'Satisfiability of ECTL* with Tree Constraints', in *Computer Science – Theory and Applications*, eds., Lev D. Beklemishev and Daniil V. Musatov, volume 9139 of *Lecture Notes in Computer Science*, 94–108, Springer International Publishing, (2015).

[6] Claudia Carapelle, Alexander Kartzow, and Markus Lohrey, 'Satisfiability of CTL* with Constraints', in *CONCUR 2013 – Concurrency Theory*, eds., Pedro R. D'Argenio and Hernán C. Melgratti, volume 8052 of *Lecture Notes in Computer Science*, pp. 455–469. Springer Berlin Heidelberg, (2013).

[7] Claudia Carapelle, Alexander Kartzow, and Markus Lohrey, 'Satisfiability of ECTL* with Constraints', *Journal of Computer and System Sciences*, (2016). To appear.

[8] Claudia Carapelle and Anni-Yasmin Turhan, 'Decidability of $\mathcal{ALC}^{\mathcal{P}}(\mathcal{D})$ for concrete domains with the EHD-property', LTCS-Report 16-01, Chair of Automata Theory, TU Dresden, (2016). See https://lat.inf.tu-dresden.de/research/reports/2016/CaTu-LTCS-16-01.pdf.

[9] Volker Haarslev, Ralf Möller, and Michael Wessel, 'The description logic $\mathcal{ALCNH}_{R+}$ extended with concrete domains: A practically motivated approach', in *In Proceedings of the First International Joint Conference on Automated Reasoning, IJCAR*, eds., R. Goré, A. Leitsch, and T. Nipkow, volume 2083 of *Lecture Notes in Computer Science*, pp. 29–44. Springer, (2001).

[10] Ian Horrocks and Ulrike Sattler, 'Ontology reasoning in the $\mathcal{SHOQ}(\mathcal{D})$ description logic', in *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI*, ed., B. Nebel, pp. 199–204, (2001).

[11] Carsten Lutz, 'Nexptime-complete description logics with concrete domains', LTCS-Report LTCS-00-01, LuFG Theoretical Computer Science, RWTH Aachen, Germany, (2000). See http://www-lti.informatik.rwth-aachen.de/Forschung/Reports.html.

[12] Carsten Lutz, 'Adding numbers to the $\mathcal{SHIQ}$ description logic—First results', in *Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR'02)*, pp. 191–202. Morgan Kaufman, (2002).

[13] Carsten Lutz, 'Description logics with concrete domains—a survey', in *Advances in Modal Logic 2002 (AiML 2002)*, Toulouse, France, (2002). Final version appeared in Advanced in Modal Logic Volume 4, 2003.

[14] Carsten Lutz, 'Combining interval-based temporal reasoning with general TBoxes', *Artificial Intelligence*, **152**(2), 235–274, (2004).

[15] Carsten Lutz, 'NExpTime-complete description logics with concrete domains', *ACM Transactions on Computational Logic*, **5**(4), 669–705, (2004).

[16] Carsten Lutz and Maja Miličić, 'A tableau algorithm for description logics with concrete domains and general TBoxes', *Journal of Automated Reasoning*, **38**(1-3), 227–259, (2007).

[17] Dorian Merz, Rafael Peñaloza, and Anni-Yasmin Turhan, 'Reasoning in $\mathcal{ALC}$ with fuzzy concrete domains', in *Proceedings of 37th edition of the German Conference on Artificial Intelligence (KI'14)*, eds., Carsten Lutz and Michael Thielscher, volume 8736 of *Lecture Notes in Artificial Intelligence*, pp. 171–182. Springer Verlag, (2014).

[18] Wolfgang Thomas, 'Languages, automata, and logic', in *Handbook of Formal Languages*, pp. 389–455. Springer, (1996).

[19] David Toman and Grant E. Weddell, 'Applications and extensions of PTIME description logics with functional constraints', in *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 948–954, (2009).

[20] W3C OWL Working Group. OWL 2 web ontology language document overview. W3C Recommendation, 27th October 2009. http://www.w3.org/TR/2009/REC-owl2-overview-20091027/.