The Abstract Expressive Power of First-Order and Description Logics with Concrete Domains

Franz Baader
TU Dresden and Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI)
Dresden, Germany
franz.baader@tu-dresden.de

Filippo De Bortoli
TU Dresden and Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI)
Dresden, Germany
filippo.de_bortoli@tu-dresden.de

ABSTRACT

Concrete domains have been introduced in description logic (DL) to enable reference to concrete objects (such as numbers) and predefined predicates on these objects (such as numerical comparisons) when defining concepts. The primary research goal in this context was to find restrictions on the concrete domain such that its integration into certain DLs preserves decidability or tractability. In this paper, we investigate the abstract expressive power of both first-order and description logics extended with concrete domains, i.e., we analyze which classes of first-order interpretations can be expressed using these logics, compared to what first-order logic without concrete domains can express. We demonstrate that, under natural conditions on the concrete domain $\mathfrak{D}$ (which also play a role for decidability), extensions of first-order logic (FOL) or the well-known DL $\mathcal{ALC}$ with $\mathfrak{D}$ share important formal characteristics with FOL, such as the compactness and the Löwenheim-Skolem properties. Nevertheless, their abstract expressive power need not be contained in that of FOL, though in some cases it is. To be more precise, we show, on the one hand, that unary concrete domains leave the abstract expressive power within FOL if we are allowed to introduce auxiliary predicates. On the other hand, we exhibit a class of concrete domains that push the abstract expressive power beyond that of FOL. As a by-product of these investigations, we obtain (semi-)decidability results for some of the logics with concrete domains considered in this paper.

CCS CONCEPTS

• Theory of computation → Description logics; **• Computing methodologies → Description logics**.

KEYWORDS

Description Logics, Concrete Domains, Model Theory, Expressive Power

ACM Reference Format:
Franz Baader and Filippo De Bortoli. 2024. The Abstract Expressive Power of First-Order and Description Logics with Concrete Domains. In *The 39th ACM/SIGAPP Symposium on Applied Computing (SAC '24), April 8–12, 2024, Avila, Spain.* ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/3605098.3635984

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SAC '24, April 8–12, 2024, Avila, Spain
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0243-3/24/04.
https://doi.org/10.1145/3605098.3635984

1 INTRODUCTION

Description Logics (DLs) [8] are a prominent family of logic-based knowledge representation formalisms, which offer a good compromise between expressiveness and the complexity of reasoning and are the formal basis for the Web ontology language OWL 2.[1] To accommodate diverse application domains, the DL community has developed logics whose expressive power is tailored towards what is needed in these domains while leaving reasoning decidable. In many cases, however, the added expressiveness turned out to be useful also in other applications. For example, concrete domains, which enable reference to concrete objects (such as numbers) and predefined predicates on these objects (such as numerical comparisons), were introduced in [6] motivated by a mechanical engineering application [7]. Due to their usefulness in many applications, they are included in the OWL 2 standard, albeit in the restricted form of unary concrete domains (called datatypes), where all predefined predicates have arity one [16].

Most DLs [8] are decidable fragments of first-order logic (FOL), i.e., their expressive power [4, 17] is below that of FOL, but there are also decidable DLs whose knowledge bases (KBs) cannot always be expressed by an FOL sentence [5]. A case in point are DLs with concrete domains [6, 18, 20], at least at first sight. In such DLs, the abstract interpretation domain is complemented by the concrete domain, and partial functions can be used to assign values in the concrete domain to abstract objects. These values can then be constrained using the predefined predicates of the concrete domain. For example, assume that we want to model physical objects, collected in a concept (i.e., unary predicate) *PO*, which can be decomposed into their proper parts using a role (i.e., binary predicate) *hpp* for "has proper part." If we want to take the weight of such objects into account, it makes sense to assign a number for its weight to every physical object using a feature (i.e., partial function) w, and to state that this weight is positive and that proper parts are physical objects that have a smaller weight than the whole. Using the syntax employed in [10, 20] and in the present paper, these conditions can be expressed with the help of value restrictions and concrete domain restrictions w.r.t. an appropriate concrete domain by the following concept inclusion (CI):

$$PO \sqsubseteq \forall hpp.\, PO \sqcap \exists w.\, (x_1 > 0) \sqcap \forall w, hpp\, w.\, {>}(x_1, x_2). \qquad (1)$$

Depending on what kind of decomposition into proper parts we have in mind, we can use the rational numbers or the integers

[1] https://www.w3.org/TR/owl2-overview/

as concrete domain. The former would be more appropriate for settings like cutting a cake, where a given piece can always be cut into even smaller parts, whereas the latter is more appropriate for settings where physical objects are composed of finitely many atomic parts that cannot be divided any further. Interestingly, as we will show in this paper, this decision also has an impact on the formal properties that the logic (in the example, the well-known DL $\mathcal{ALC}$ [26]) extended with such a concrete domain satisfies. If we employ the integers, then for any element of *PO* there is a positive integer such that the length of all *hpp*-chains issuing from it are bounded by this number. Using this fact, it is easy to show that the logic at hand is not compact, i.e., there may be unsatisfiable infinite sets of sentences for which all finite subsets are satisfiable. In particular, this implies that the abstract expressive power of this logic, which considers only the abstract domain and the interpretation of concept and role names, but ignores the feature values, cannot be contained in FOL. For the rational numbers, the results obtained in this paper imply that the extension of $\mathcal{ALC}$ or FOL with this concrete domain shares the compactness and the Löwenheim-Skolem property with FOL. The reason is that the rational numbers with > are *homomorphism* ω*-compact* [9, 10], which means that a countable set of constraints is solvable iff all its finite subsets are solvable. We can, however, prove that the abstract expressive power of these logics is nevertheless not contained in FOL, though we cannot use a compactness argument to show this.

Note that it is quite natural to consider concrete domains that are homomorphism ω-compact. In fact, in the presence of CIs, integrating even rather simple concrete domains into the DL $\mathcal{ALC}$ may cause undecidability [9, 19]. To overcome this problem, the notion of ω*-admissible* concrete domains was introduced in [20], and it was proved that integrating such a concrete domain into $\mathcal{ALC}$ leaves reasoning decidable also in the presence of CIs. By definition, all ω-admissible concrete domains are homomorphism ω-compact. As examples of ω-admissible concrete domains, Allen's interval algebra [1] and the region connection calculus RCC8 [24] are provided in [20]. Using well-known notions and results from model theory, additional ω-admissible concrete domains were exhibited in [9, 10]. A simpler, but considerably more restrictive way of achieving decidability is to use *unary* concrete domains. Decidability for the expressive DL $\mathcal{SHOQ}$ extended with such concrete domains is shown in [16]. In [3, 25], conjunctive query answering in extensions of the inexpressive DL DL-Lite with unary concrete domains is investigated.

In the next section, we will introduce FOL with concrete domains and then define DLs with concrete domains as fragments. Subsequently, we present two variants of the notion of abstract expressive power, one where one can use auxiliary predicates on the first-order side to express sentences of the logic with concrete domains, and one where this is not allowed. Section 3 is dedicate to proving that FOL and DLs with concrete domains share a number of interesting formal properties with FOL, provided that the employed concrete domain is homomorphism ω-compact and its set of predicates is closed under negation. In Section 4, we show, on the one hand, that FOL with a unary concrete domain can be expressed in FOL if we are allowed to use auxiliary predicates. In addition, if we restrict the logic with unary concrete domain to a decidable fragment like the guarded or the two-variable fragment with counting, then decidability on the concrete domain side yields decidability of the whole logic. On the other hand, we provide conditions on concrete domains such that $\mathcal{ALC}$ extended with such a concrete domain cannot be expressed in FOL. Basically, these are concrete domains whose predicates are closed under negation and in which equality is definable.

2 LOGICS WITH CONCRETE DOMAINS

We introduce first-order logic with concrete domains, from which we obtain DLs with concrete domains as fragments. Then, we define the notion of abstract expressive power of a logic with concrete domains. We assume that the reader is familiar with syntax, semantics, and basic results for first-order logic.

Concrete domains. A concrete domain is a τ-structure $\mathfrak{D}$ for a relational signature τ, i.e., it consists of a set D, called its *domain*, together with relations $P^D \subseteq D^k$ for each k-ary relation symbol $P \in \tau$. A *constraint system* Γ for $\mathfrak{D}$ is a set of atoms of the form $P(x_1, \dots, x_k)$ where $P \in \tau$ has arity k and the x_i are variables. The constraint system Γ is *satisfiable* in $\mathfrak{D}$ if there is an assignment h (also called *homomorphism*) of elements of D to the variables in Γ such that $(h(x_1), \dots, h(x_k)) \in P^D$ for all atoms $P(x_1, \dots, x_k)$ in Γ. We call such a homomorphism a *solution* of Γ in $\mathfrak{D}$. For example, consider the structure $\mathfrak{Q} := (\mathbb{Q}, >)$ of rational numbers with the standard ordering relation, which we write infix. The set $\Gamma := \{x_1 > x_2, x_2 > x_3, x_3 > x_1\}$ is a finite constraint system that is *unsatisfiable* (i.e., not satisfiable) in $\mathfrak{Q}$.

The concrete domain $\mathfrak{D}$ is *homomorphism* ω*-compact* if the following holds for any countable constraint system Γ for $\mathfrak{D}$: Γ is satisfiable in $\mathfrak{D}$ iff all its finite subsets are satisfiable in $\mathfrak{D}$. For example, $\mathfrak{Q}$ is homomorphism ω-compact. This follows from the results in [9, 10], but is also a consequence of the fact that a constraint system is satisfiable in $\mathfrak{Q}$ iff it does not contain a cycle of the form $x_1 > x_2, x_2 > x_3, \dots, x_n > x_1$. In contrast $\mathfrak{Z} := (\mathbb{Z}, >)$ is not homomorphism ω-compact: the constraint system $\Gamma := \{x_q > x_r \mid q, r \in \mathbb{Q}, q > r\}$ is countable and unsatisfiable in $\mathfrak{Z}$ since it requires the existence of infinitely many integers between whatever integers are assigned to x_0 and x_1; however, any finite subset is clearly satisfiable.

First-order logic with concrete domains. Let $\mathfrak{D}$ be a concrete domain over a relational signature τ, σ be a first-order signature (which may also contain function symbols), and $\mathcal{F}$ be a countable set of *feature symbols*. The formulae of *first-order logic with the concrete domain* $\mathfrak{D}$, $\mathsf{FOL}_{\sigma}^{\mathcal{F}}(\mathfrak{D})$ (or simply $\mathsf{FOL}(\mathfrak{D})$ if σ and $\mathcal{F}$ are irrelevant or clear from the context), are obtained by extending the usual inductive definition for FOL with the following two base cases:

- *definedness predicates* $\mathrm{Def}(f)(t)$ with $f \in \mathcal{F}$ and t a σ-term,
- *concrete domain predicates* $P(f_1, \dots, f_n)(t_1, \dots, t_n)$ with $P \in \tau$ of arity n, $f_i \in \mathcal{F}$, and t_i σ-terms.

The semantics of $\mathsf{FOL}(\mathfrak{D})$ formulae is defined inductively, using a first-order interpretation $\mathfrak{I} = (I, \cdot^{\mathfrak{I}})$ for σ extended with a set $\mathfrak{F}$ of partial functions $f^{\mathfrak{F}}\colon I \rightharpoonup D$ for $f \in \mathcal{F}$, and an assignment w mapping variables to elements of I. The semantics of terms, Boolean connectives and first-order quantifiers is defined as usual, where we denote the interpretation of a term t by $\mathfrak{I}$ and w as $t^{\mathfrak{I},w}$. The new predicates are interpreted as follows, where $\bar{f} := f_1, \ldots, f_n$ and $\bar{t} := t_1, \ldots, t_n$:

- $(\mathfrak{I}, \mathfrak{F}), w \models \mathrm{Def}(f)(t)$ if $f^{\mathfrak{F}}(t^{\mathfrak{I},w})$ is defined, and
- $(\mathfrak{I}, \mathfrak{F}), w \models P(\bar{f})(\bar{t})$ if $(f_1^{\mathfrak{F}}(t_1^{\mathfrak{I},w}), \ldots, f_n^{\mathfrak{F}}(t_n^{\mathfrak{I},w})) \in P^D$.

Note that $(f_1^{\mathfrak{F}}(t_1^{\mathfrak{I},w}), \ldots, f_n^{\mathfrak{F}}(t_n^{\mathfrak{I},w})) \in P^D$ entails that $f_i^{\mathfrak{F}}(t_i^{\mathfrak{I},w})$ must be defined for $i = 1, \ldots, n$. The tuple $(\mathfrak{I}, \mathfrak{F})$ is a *model* of the $\mathtt{FOL}(\mathfrak{D})$ sentence ϕ (i.e., formula without free variables), in symbols $(\mathfrak{I}, \mathfrak{F}) \models \phi$, if $(\mathfrak{I}, \mathfrak{F}), w \models \phi$ for some (and thus all) assignments w.

Description Logics with concrete domains. For an arbitrary DL $\mathcal{DL}$, a given concrete domain $\mathfrak{D}$ can be integrated into $\mathcal{DL}$ with the help of concrete domain restrictions. *Concrete domain restrictions* for $\mathfrak{D}$ are concept constructors of the form $\exists \bar{p}.P(\bar{x})$ or $\forall \bar{p}.P(\bar{x})$, with $\bar{p} = p_1, \ldots, p_k$ a sequence of k feature paths, P a k-ary predicate of $\mathfrak{D}$, and $\bar{x} = x_1, \ldots, x_k$ a k-tuple of variables. In the context of this paper, a *feature path* is either a feature name f or an expression rf with f a feature name and r a role name. We denote the DL obtained from $\mathcal{DL}$ by adding these restrictions as concept constructors with $\mathcal{DL}(\mathfrak{D})$. For example, $\mathcal{ALC}(\mathfrak{D})$ has, in addition to the concrete domain restrictions introduced above, the concept constructors conjunction ($\sqcap$), disjunction ($\sqcup$), negation ($\neg$), existential restriction ($\exists r.C$), and value restriction ($\forall r.C$).

To define the semantics of $\mathcal{DL}(\mathfrak{D})$, we assume that concepts of $\mathcal{DL}$ can inductively be translated into $\mathtt{FOL}$ formulae with one free variable x using a translation function π_x. For example, $\pi_x(C \sqcap D) := \pi_x(C) \wedge \pi_x(D)$ and $\pi_x(\exists r.C) := \exists y.(r(x,y) \wedge \pi_y(C))$. We extend this translation function to map concepts of $\mathcal{DL}(\mathfrak{D})$ to formulae of $\mathtt{FOL}(\mathfrak{D})$ by providing the translation of concrete domain restrictions. Taking $\bar{x}, \bar{p}$ as defined above, let $I \subseteq \{1, \ldots, k\}$ be such that $p_i = r_i f_i$ if $i \in I$ and $p_i = f_i$ otherwise. We define $\bar{y} := y_1, \ldots, y_k$ by setting $y_i = x_i$ if $i \in I$ and $y_i = x$ otherwise, and $\bar{z}$ as the sequence of variables y_i with $i \in I$. The translation of concrete domain restrictions is then defined as follows, where $\gamma(x, \bar{y})$ abbreviates $\bigwedge_{i \in I} r_i(x, y_i) \wedge \bigwedge_{i=1}^{k} \mathrm{Def}(f_i)(y_i)$:

$$\begin{aligned} \pi_x(\exists \bar{p}.P(\bar{x})) &:= \exists \bar{z}.\, (\textstyle\bigwedge_{i \in I} r_i(x, y_i) \wedge P(f_1, \ldots, f_k)(\bar{y})), \\ \pi_x(\forall \bar{p}.P(\bar{x})) &:= \forall \bar{z}.\, (\gamma(x, \bar{y}) \rightarrow P(f_1, \ldots, f_k)(\bar{y})). \end{aligned} \tag{2}$$

The semantics of TBoxes (i.e., finite sets of CIs $C \sqsubseteq D$) of the DL $\mathcal{DL}(\mathfrak{D})$ is then defined in the usual way by translation into $\mathtt{FOL}(\mathfrak{D})$ sentences: $C \sqsubseteq D$ is translated into $\forall x.\pi_x(C) \rightarrow \pi_x(D)$. It is easy to see that the semantics of concrete domain restrictions given by the translation in (2) coincides with the direct model-theoretic semantics in [10, 20]. In [20], extensions of the predicates of a concrete domain $\mathfrak{D}$ by disjunctions of its base predicates are allowed to be used in concrete domain restrictions, whereas in [10] even predicates first-order definable from the base predicates are considered. These extensions can clearly also be translated into $\mathtt{FOL}(\mathfrak{D})$. We denote them as $\mathcal{DL}_{\vee^+}(\mathfrak{D})$ and $\mathcal{DL}_{\mathrm{fo}}(\mathfrak{D})$, respectively.

Abstract expressive power. If we want to compare the expressive power of (a fragment of) $\mathtt{FOL}$ with that of (a fragment of) $\mathtt{FOL}(\mathfrak{D})$, we have the problem that the semantic structures they are based on differ in that, for the latter, one additionally has a collection of partial functions into the concrete domain. To overcome this difference, we say that the first-order interpretation $\mathfrak{I}$ is an *abstract model* of the $\mathtt{FOL}(\mathfrak{D})$ sentence ϕ, in symbols $\mathfrak{I} \models_{\mathfrak{D}} \phi$, if there is an interpretation of the feature symbols $\mathfrak{F}$ such that $(\mathfrak{I}, \mathfrak{F}) \models \phi$. The $\mathtt{FOL}$ sentence ψ is an *abstract definition* of the $\mathtt{FOL}(\mathfrak{D})$ sentence ϕ if the abstract models of ϕ are exactly the models of ψ. In this case we also say that ϕ and ψ are *abstractly equivalent.*

Example 2.1. Consider the unary concrete domain $\mathfrak{N} := (\mathbb{N}, \mathsf{even}, \mathsf{odd})$ where $\mathsf{even}, \mathsf{odd}$ are unary relations with the standard meaning. The $\mathcal{ALC}(\mathfrak{N})$ TBox $\mathcal{T} := \{A \sqsubseteq \exists f.\mathsf{even}(x), B \sqsubseteq \exists f.\mathsf{odd}(x)\}$ is abstractly equivalent to the $\mathcal{ALC}$ TBox $\mathcal{T}' := \{A \sqsubseteq \neg B\}$. In fact, A and B must be interpreted as disjoint sets in any model of $\mathcal{T}$. Conversely, any model of $\mathcal{T}'$ can be extended to a model of $\mathcal{T}$ by defining f to yield 0 for the elements of A, 1 for the elements of B, and no value for all other elements.

We will show in the next section that such a definability result always holds for unary concrete domains. However, in general one may need to introduce auxiliary predicates to express the concrete domain restrictions. The following definition allows for such additional predicates. Let ϕ be an $\mathtt{FOL}(\mathfrak{D})$ sentence and ψ an $\mathtt{FOL}$ sentence that may contain auxiliary predicates not occurring in ϕ. Then ψ is an *abstract projective definition* of ϕ if the abstract models of ϕ are exactly the reducts of the models of ψ, where in a reduct we just forget about the interpretation of the auxiliary predicates. In this case we also say that ϕ and ψ are *abstractly projectively equivalent.* The abstract expressive power of (a fragment of) $\mathtt{FOL}(\mathfrak{D})$ is determined by which classes of abstract models can be defined by its sentences.

Definition 2.2. Given a fragment F of $\mathtt{FOL}(\mathfrak{D})$, we say that its *abstract expressive power* is *(projectively) contained* in a fragment G of $\mathtt{FOL}$ if every sentence of F has an abstract (projective) definition in G.

Example 2.3. In the introduction we have given an example showing that, for a concrete domain $\mathfrak{D}$ over the integers with predicates $x > y$ and $x > 0$, the abstract expressive power of $\mathcal{ALC}(\mathfrak{D})$ is not contained in $\mathtt{FOL}$. The argument we have used there (which is based on the fact that $\mathtt{FOL}$ is compact, but $\mathcal{ALC}(\mathfrak{D})$ is not) also works in the projective setting. In fact, the CI (1) enforces that, for any element of *PO*, there is a positive integer such that the length of all *hpp*-chains issuing from it are bounded by this number. Assume that ψ is an $\mathtt{FOL}$ sentence that is an abstract projective definition of this CI. Clearly we can write, for all $n \geq 1$, an $\mathtt{FOL}$ sentence ψ_n that says that the constant a is an element of *PO* and the starting point of an *hpp*-chain of length n. Then any finite subset of $\{\psi\} \cup \{\psi_n \mid n \geq 1\}$ is satisfiable, but the whole set cannot be satisfiable since the CI (1) enforces a finite bound on the length of chains issuing from a. Since $\mathtt{FOL}$ is compact, this shows that ψ cannot be a first-order sentence.

However, compactness of $\mathcal{ALC}(\mathfrak{D})$ for a given concrete domain $\mathfrak{D}$ does not guarantee that its abstract expressive power is projectively contained in $\mathtt{FOL}$.

Example 2.4. Consider the concrete domain $\mathfrak{Q}' := (\mathbb{Q}, >, =)$. The results shown in the next section imply that the logic $\mathtt{FOL}(\mathfrak{Q}')$ is compact, and thus also its fragment $\mathcal{ALC}(\mathfrak{Q}')$. Nevertheless, the abstract expressive power of $\mathcal{ALC}(\mathfrak{Q}')$ is not projectively contained in $\mathtt{FOL}$. To see this, consider the TBox $\mathcal{T} := \{\top \sqsubseteq \exists f, f.{=}(x_1, x_2) \sqcap \forall f, rf.{>}(x_1, x_2)\}$ and assume that there is an $\mathtt{FOL}$ formula ψ that is an abstract projective definition of it. Then $(\mathbb{Q}, >)$, where $>$ is the interpretation of r, is an abstract model of $\mathcal{T}$. In fact, one can use the identity function to interpret the feature f. Thus, $(\mathbb{Q}, >)$ can be extended to a model of ψ (by appropriate interpretations of the auxiliary predicates contained in ψ, if any). Since $(\mathbb{Q}, >)$ satisfies the formula $\tau := \forall x, y.(r(x, y) \vee x = y \vee r(y, x))$, we can conclude that $\psi \wedge \tau$ is satisfiable. The upward Löwenheim-Skolem property of FOL yields an uncountable model of $\psi \wedge \tau$. Since ψ is an abstract projective definition of $\mathcal{T}$, the reduct $\mathfrak{R}$ of this uncountable model to the signature consisting of r must be extendable to a model of $\mathcal{T}$. This means that there is an interpretation $f^{\mathfrak{F}}$ of f such that $(\mathfrak{R}, \mathfrak{F})$ is a model of $\mathcal{T}$. The conjunct $\exists f, f.{=}(x_1, x_2)$ on the right-hand side of the CI forces $f^{\mathfrak{F}}$ to be total. Let ν, μ be distinct elements of $\mathfrak{R}$. Since $\mathfrak{R}$ satisfies τ, we know that ν and μ are related via r, in one direction or the other. Then the restriction $\forall f, rf.{>}(x_1, x_2)$ yields $f^{\mathfrak{F}}(\nu) \neq f^{\mathfrak{F}}(\mu)$, and thus $f^{\mathfrak{F}}$ is injective. However, since $\mathfrak{R}$ is uncountable and $\mathbb{Q}$ is countable, there cannot be an injective function from the domain of $\mathfrak{R}$ to $\mathbb{Q}$.

3 FIRST-ORDER PROPERTIES OF LOGICS WITH CONCRETE DOMAINS

First-order logic satisfies a number of interesting formal characteristics, usually shown in any introductory textbook in logic [12, 14]. Assuming that Φ is an at most countable set of sentences in our target language, these properties can be specified as follows:

(Downward) Löwenheim-Skolem: If Φ is satisfiable, then it has a model whose domain is at most countable;

(Upward) Löwenheim-Skolem: If Φ has a model with an infinite domain, then it has a model with an uncountable domain;

(Countable) Compactness: If every finite subset of Φ is satisfiable, then Φ is satisfiable;

Recursive enumerability: The set of unsatisfiable sentences is recursively enumerable (r.e.).

We will show that, under natural conditions on the concrete domain $\mathfrak{D}$, $\mathsf{FOL}(\mathfrak{D})$ shares most and $\mathcal{ALC}(\mathfrak{D})$ shares all of these properties with FOL. The first condition is that $\mathfrak{D}$ is *homomorphism ω-compact*, i.e., that constraint solving in $\mathfrak{D}$ is compact in the sense that a countable constraint system for $\mathfrak{D}$ is satisfiable iff every of its finite subsets is satisfiable. As mentioned before, this property is part of the ω-admissibility condition, which guarantees decidability of $\mathcal{ALC}(\mathfrak{D})$. The second condition is that the concrete domain $\mathfrak{D}$ is *closed under negation*, i.e. for every predicate symbol P of $\mathfrak{D}$ there is a predicate symbol P_c of $\mathfrak{D}$ such that $\bar{d} \in P^D$ iff $\bar{d} \notin P_c^D$. This condition appears in the definition of *admissibility* for concrete domains [6], and is needed since our logics can express negation of concrete domain predicates. If it is not satisfied (as, e.g., for the concrete domain $\mathfrak{Q}'$ in Example 2.4), then one can extend the concrete domain by the missing predicates. However, then homomorphism ω-compactness needs to hold for the extended concrete domain (as is the case for the extension of $\mathfrak{Q}'$ by the complements of its predicates).

We assume in this section that the concrete domain $\mathfrak{D}$ is homomorphism ω-compact and closed under negation. The main tool for showing our results is a satisfiability-preserving translation of sets of $\mathsf{FOL}(\mathfrak{D})$ sentences into sets of FOL sentences.

Rewriting to first-order logic. Let Φ be an at most countable set of $\mathsf{FOL}(\mathfrak{D})$ sentences. We translate Φ into a set of FOL sentences Φ^{FOL} by replacing every atom of the form $P(f_1, \ldots, f_n)(t_1, \ldots, t_n)$ occurring in Φ with $P^{f_1,\ldots,f_n}(t_1, \ldots, t_n)$, where for every n-ary concrete domain predicate P and features $f_1, \ldots, f_n$ we assume that $P^{f_1,\ldots,f_n}$ is a new n-ary predicate symbol in the first-order signature. Similarly, every atom of the form $\mathrm{Def}(f)(t)$ is replaced with $\mathrm{Def}_f(t)$ where Def_f is a new predicate symbol for every feature f.

Every set Γ of atoms of the form $P^{f_1,\ldots,f_n}(x_1, \ldots, x_n)$ induces the constraint system $\widehat{\Gamma} := \{P(f^1_{x_1}, \ldots, f^n_{x_n}) \mid P^{f^1,\ldots,f^n}(x_1, \ldots, x_n) \in \Gamma\}$, where f_x is a new variable for each feature name f and variable x. To capture the semantics of the concrete domain predicates and the definedness predicate, we additionally consider the set of FOL sentences $\Psi^{\mathfrak{D}}$ where:

- assuming that $\bar{x} := x_1, \ldots, x_n$, we add for each of the new predicate symbols $P^{f_1,\ldots,f_n}$ the sentences

$$\forall \bar{x}.P^{f_1,\ldots,f_n}(\bar{x}) \to \mathrm{Def}_{f_1}(x_1) \wedge \cdots \wedge \mathrm{Def}_{f_n}(x_n),$$
$$\forall \bar{x}.\neg P^{f_1,\ldots,f_n}(\bar{x}) \to P_c^{f_1,\ldots,f_n}(\bar{x}) \vee \textstyle\bigvee_{i=1}^{n}(\neg \mathrm{Def}_{f_i}(x_i)),$$

- for every finite set Γ of atoms of the form $P^{f_1,\ldots,f_n}(x_1, \ldots, x_n)$ we add the sentence $\forall \bar{x}. \bigwedge \Gamma \to \bot$ if the constraint system $\widehat{\Gamma}$ is unsatisfiable in $\mathfrak{D}$, where $\bar{x}$ collects all the variables occurring in Γ.

Theorem 3.1. *Let $\mathfrak{D}$ be a homomorphism ω-compact concrete domain that is closed under negation. The set Φ of $\mathsf{FOL}(\mathfrak{D})$ formulae is satisfiable in $\mathsf{FOL}(\mathfrak{D})$ iff $\Phi^{\mathsf{FOL}} \cup \Psi^{\mathfrak{D}}$ is satisfiable in FOL.*

Proof. "$\Leftarrow$" Assume that $\Phi^{\mathsf{FOL}} \cup \Psi^{\mathfrak{D}}$ is satisfiable. Since this is a countable set of first-order formulae, we apply the downward Löwenheim-Skolem property of FOL to get an at most countable model $\mathfrak{I}$ of $\Phi^{\mathsf{FOL}} \cup \Psi^{\mathfrak{D}}$. We show that we can extend $\mathfrak{I}$ with an interpretation $\mathfrak{F}$ of the features such that $(\mathfrak{I}, \mathfrak{F})$ is a model of Φ. To this purpose, introduce a fresh variable x_d for every $d \in I$ and consider the set $\Gamma_{\mathfrak{I}}$ consisting of all atoms $P^{f^1,\ldots,f^n}(x_{d_1}, \ldots, x_{d_n})$ such that $P^{f^1,\ldots,f^n}(d_1, \ldots, d_n)$ is satisfied in $\mathfrak{I}$, where $d_1, \ldots, d_n$ ranges over all elements of $\mathfrak{I}$ and $f^1, \ldots, f^n$ over all feature names. Due to our construction of $\Psi^{\mathfrak{D}}$ and the fact that $\mathfrak{I}$ is a model of this set, we know that all finite subsets of $\widehat{\Gamma}_{\mathfrak{I}}$ are satisfiable in $\mathfrak{D}$. Since $\widehat{\Gamma}_{\mathfrak{I}}$ is countable, homomorphism ω-compactness implies that there exists a solution h of it in $\mathfrak{D}$. For all feature names f and elements $d \in I$ for which the variable f_{x_d} occurs in $\widehat{\Gamma}_{\mathfrak{I}}$, we define $f^{\mathfrak{F}}(d) := h(f_{x_d})$. Otherwise, we choose an arbitrary value for $f^{\mathfrak{F}}(d)$ if $\mathrm{Def}_f(d)$ is true in $\mathfrak{I}$, and leave $f^{\mathfrak{F}}(d)$ undefined otherwise. The fact that, together with this interpretation of the features $\mathfrak{F}$, the FOL interpretation $\mathfrak{I}$ is indeed a model of Φ, is an immediate consequence of the following two claims:

(1) $\mathrm{Def}_f(d)$ is true in $\mathfrak{I}$ iff $\mathrm{Def}(f)(d)$ is true in $(\mathfrak{I}, \mathfrak{F})$;
(2) $P^{f_1,\ldots,f_n}(d_1, \ldots, d_n)$ is true in $\mathfrak{I}$ iff $P(f_1, \ldots, f_n)(d_1, \ldots, d_n)$ is true in $(\mathfrak{I}, \mathfrak{F})$.

To show the first claim, assume that $\mathrm{Def}_f(d)$ is true in $\mathfrak{I}$. Then $f^{\mathfrak{F}}(d)$ is defined either by the solution h of the constraint system $\widehat{\Gamma}_{\mathfrak{I}}$ in $\mathfrak{D}$ or it has received some arbitrary value. If $\mathrm{Def}_f(d)$ is not true in $\mathfrak{I}$, then $f^{\mathfrak{F}}(d)$ cannot have been defined in terms of h, since otherwise an expression $P^{f_1,\dots,f_n}(d_1,\dots,d_n)$ that is true in $\mathfrak{I}$ would have to exist such that $f = f_i$ and $d = d_i$. But then $\Psi^{\mathfrak{D}}$ would have enforced $\mathrm{Def}_f(d)$ to be true in $\mathfrak{I}$, leading to a contradiction. In addition, since $\mathrm{Def}_f(d)$ is not true in $\mathfrak{I}$, no arbitrary value is assigned to $f^{\mathfrak{F}}(d)$. Thus $f^{\mathfrak{F}}(d)$ is undefined.

Regarding the second claim, first assume $P^{f_1,\dots,f_n}(d_1,\dots,d_n)$ is true in $\mathfrak{I}$, which means that $P^{f^1,\dots,f^n}(x_{d_1},\dots,x_{d_n})$ belongs to $\Gamma_{\mathfrak{I}}$. Since $\mathfrak{F}$ was defined using a solution of $\widehat{\Gamma}_{\mathfrak{I}}$, we know that $P(f_1^{\mathfrak{F}}(d_1),\dots,f_n^{\mathfrak{F}}(d_n))$ holds in $\mathfrak{D}$, and thus $P(f_1,\dots,f_n)(d_1,\dots,d_n)$ is true in $(\mathfrak{I},\mathfrak{F})$. Conversely, assume that $P^{f_1,\dots,f_n}(d_1,\dots,d_n)$ is not true in $\mathfrak{I}$, which means that its negation is true in $\mathfrak{I}$. Since $\mathfrak{I}$ is a model of $\Psi^{\mathfrak{D}}$, this implies that

$$P_c^{f_1,\dots,f_n}(d_1,\dots,d_n) \vee \neg\,\mathrm{Def}_{f_1}(d_1) \vee \dots \vee \neg\,\mathrm{Def}_{f_n}(d_n)$$

is true in $\mathfrak{I}$. If $P_c^{f_1,\dots,f_n}(d_1,\dots,d_n)$ is true in $\mathfrak{I}$, then we can employ the same approach as for the only-if direction to show that $P_c(f_1,\dots,f_n)(d_1,\dots,d_n)$ is true in $(\mathfrak{I},\mathfrak{F})$, which clearly implies that $P(f_1,\dots,f_n)(d_1,\dots,d_n)$ cannot be true in $(\mathfrak{I},\mathfrak{F})$. Similarly, if $\neg\,\mathrm{Def}_{f_i}(d_i)$ is true in $\mathfrak{I}$, then according to the first claim, $\mathrm{Def}(f_i)(d_i)$ cannot be true in $(\mathfrak{I},\mathfrak{F})$. Thus, $P(f_1,\dots,f_n)(d_1,\dots,d_n)$ also cannot be true since $\mathfrak{I}$ satisfies $\Psi^{\mathfrak{D}}$.

"$\Rightarrow$" Assume that Φ is satisfiable in $\mathsf{FOL}(\mathfrak{D})$ by the interpretation $\mathfrak{I}$ of the FOL part and the interpretation $\mathfrak{F}$ of the features. We extend $\mathfrak{I}$ to an interpretation $\mathfrak{I}'$ that also takes the new predicates Def_f and $P^{f_1,\dots,f_n}$ into account:

- $d \in \mathrm{Def}_f^{\mathfrak{I}'}$ iff $f^{\mathfrak{F}}(d)$ is defined,
- $(d_1,\dots,d_n) \in (P^{f_1,\dots,f_n})^{\mathfrak{I}'}$ iff $(f_1^{\mathfrak{F}}(d_1),\dots,f_n^{\mathfrak{F}}(d_n)) \in P^{\mathfrak{D}}$.

Since $(\mathfrak{I},\mathfrak{F})$ makes Φ true, it is easy to see that $\mathfrak{I}'$ is a model of Φ^{FOL}. In addition, it is a model of $\Psi^{\mathfrak{D}}$ due to the semantics of concrete domain restriction in $\mathsf{FOL}(\mathfrak{D})$ and the fact that P_c is the complement of P in $\mathfrak{D}$. ◻

Thanks to this theorem, we can transfer the properties of FOL introduced above to $\mathsf{FOL}(\mathfrak{D})$.

Corollary 3.2. *FODcompactness If $\mathfrak{D}$ is a homomorphism ω-compact concrete domain that is closed under negation, then $\mathsf{FOL}(\mathfrak{D})$ is countably compact and satisfies the downward Löwenheim-Skolem property. Homomorphism ω-compactness is also a necessary condition for countable compactness. In general, $\mathsf{FOL}(\mathfrak{D})$ need not satisfy the upward Löwenheim-Skolem property. If the finite unsatisfiable constraint systems for $\mathfrak{D}$ are r.e., then so are the unsatisfiable sentences of $\mathsf{FOL}(\mathfrak{D})$.*

Proof sketch. Compactness follows from Theorem 3.1. In fact, if Φ is unsatisfiable, then this theorem and compactness of FOL yield a finite subset Ψ of $\Phi^{\mathsf{FOL}} \cup \Psi^{\mathfrak{D}}$ that is unsatisfiable. Then translating $\Psi \cap \Phi^{\mathsf{FOL}}$ back to $\mathsf{FOL}(\mathfrak{D})$ yields an unsatisfiable finite subset of Φ. The downward Löwenheim-Skolem property follows from the construction of the abstract model $\mathfrak{I}$ in the if-direction of Theorem 3.1, which is at most countable.

Assume that the countable constraint system Γ for $\mathfrak{D}$ is a counterexample to the homomorphism ω-compactness of $\mathfrak{D}$. Then

$$\Phi_\Gamma := \{\forall x.(P(f_{x_1},\dots,f_{x_n})(x,\dots,x)) \mid P(x_1,\dots,x_n) \in \Gamma\}$$

is a countable set of $\mathsf{FOL}(\mathfrak{D})$ sentences that is a counterexample to countable compactness of $\mathsf{FOL}(\mathfrak{D})$.

Next, consider the concrete domain $\mathfrak{Q}_= := (\mathbb{Q}, =, \neq)$, which is closed under negation and easily seen to be homomorphism ω-compact. The $\mathsf{FOL}(\mathfrak{Q}_=)$ sentence

$$\phi_{up} := \forall x, y.\, \mathrm{Def}(f)(x) \wedge (x \neq y \rightarrow \neq(f,f)(x,y))$$

states that f is an injective function from the domain of an abstract model of ϕ_{up} into $\mathbb{Q}$. Thus, no abstract model of ϕ_{up} can have an uncountable domain, as $\mathbb{Q}$ is is countable.

Finally, assume that $\Phi = \{\phi\}$ for an $\mathsf{FOL}(\mathfrak{D})$ sentence ϕ. The assumption that the finite unsatisfiable constraint systems for $\mathfrak{D}$ are r.e. entails that the set $\Phi^{\mathsf{FOL}} \cup \Psi^{\mathfrak{D}}$ is r.e. as well. We can now dovetail a partial decision procedure for unsatisfiability of finite sets of FOL sentences with the enumeration of $\Phi^{\mathsf{FOL}} \cup \Psi^{\mathfrak{D}}$ to get a procedure that terminates iff $\Phi^{\mathsf{FOL}} \cup \Psi^{\mathfrak{D}}$ is unsatisfiable. Together with Theorem 3.1 this shows that unsatisfiability of $\mathsf{FOL}(\mathfrak{D})$ sentences is partially decidable, and thus r.e. ◻

For $\mathcal{ALC}$ with a concrete domain, we can strengthen the result of Corollary 3.2 as following.

Corollary 3.3. *ALCDcompactness Let $\mathfrak{D}$ be a homomorphism ω-compact concrete domain that is closed under negation, and $\mathcal{L}$ be either $\mathcal{ALC}(\mathfrak{D})$, $\mathcal{ALC}_{\vee^+}(\mathfrak{D})$ or $\mathcal{ALC}_{fo}(\mathfrak{D})$. Then $\mathcal{L}$ is countably compact and satisfies the upward and the downward Löwenheim-Skolem property. Homomorphism ω-compactness is also a necessary condition for countable compactness.*

Proof sketch. The downward Löwenheim-Skolem property and countable compactness are an immediate consequence of the fact that $\mathcal{L}$ can be expressed in $\mathsf{FOL}(\mathfrak{D})$. Regarding necessity of homomorphism ω-compactness, it is easy to see that a counterexample to this property for $\mathfrak{D}$ can also be turned into a counterexample to countable compactness of $\mathcal{L}$, similar to the construction for $\mathsf{FOL}(\mathfrak{D})$. The upward Löwenheim-Skolem property is an immediate consequence of the fact that, like $\mathcal{ALC}$ [8], its extension $\mathcal{L}$ is closed under disjoint unions. ◻

4 FIRST-ORDER (NON-)DEFINABILITY AND DECIDABILITY

In Section 2, we have seen an example of a unary concrete domain $\mathfrak{R}$ and an $\mathcal{ALC}(\mathfrak{R})$ TBox $\mathcal{T}$ such that $\mathcal{T}$ is abstractly equivalent to an $\mathcal{ALC}$ TBox $\mathcal{T}'$. Basically, the first part of this section generalizes this result to all unary concrete domains $\mathfrak{D}$ that are closed under negation. To be more precise, we show that, in this setting, every $\mathsf{FOL}(\mathfrak{D})$ sentence has an abstract projective definition in FOL, and likewise every $\mathcal{ALC}(\mathfrak{D})$ TBox is abstractly projectively equivalent to an $\mathcal{ALC}$ TBox. As a byproduct of these results, we are able to show that, under the additional assumption that constraint satisfaction in $\mathfrak{D}$ is decidable, extending the guarded or two-variable fragments with counting of FOL with such a concrete domain leaves the resulting logic decidable.

Regarding non-definability, Section 2 presents an example of a homomorphism ω-compact concrete domain $\mathfrak{Q}'$ and an $\mathcal{ALC}(\mathfrak{Q}')$ TBox $\mathcal{T}$ that has no abstract projective definition in FOL. In the second part of this section, we will generalize this result from $\mathfrak{Q}'$ to countable concrete domains in which (in)equality is appropriately definable. We will also show that adding such concrete domains to FOL destroys the upward Löwenheim-Skolem property.

4.1 Unary concrete domains

We recall that a concrete domain is *unary* if it contains only unary relations. Assume that $\mathfrak{D}$ is a unary concrete domain that is closed under negation. Let ϕ be an $\mathsf{FOL}(\mathfrak{D})$ sentence and $\Phi := \{\phi\}$. The rewriting approach described in Section 3 produces a singleton set Φ^{FOL} consisting of an FOL sentence ϕ^{FOL} and a set $\Psi^{\mathfrak{D}}$ of FOL sentences consisting of

- finitely many sentences $\forall x.P^f(x) \to \mathrm{Def}_f(x)$,
- finitely many sentences $\forall x.\neg P^f(x) \to P^f_c(c) \vee \neg\,\mathrm{Def}_f(x)$,
- finitely many sentences of the form $\forall x.\Gamma \to \bot$ where Γ is a set of atoms $\{P^f_1(x), \dots, P^f_n(x)\}$ s.t. $\hat{\Gamma}$ is unsatisfiable in $\mathfrak{D}$.

The first two points are justified by the fact that we can restrict our attention to the concrete domain predicates and feature symbols that occur in ϕ. Regarding the last point, we notice that, in a setting where all concrete domain predicates are unary, constraints of the form $P^f(x)$ and $Q^g(y)$, where $f \neq g$ or $x \neq y$, cannot influence each other. Thus, one can restrict the attention to constraint systems built using a single feature name f and variable x. In fact, any unsatisfiable constraint systems must contain an unsatisfiable one of this form. Since we can again restrict the attention to the concrete domain predicates and feature symbols occurring in ϕ, and the name of single variable is irrelevant, there are only finitely many sentences of this form. Overall, this rewriting approach yields an FOL sentence $\psi := \phi^{\mathsf{FOL}} \wedge \bigwedge \Psi^{\mathfrak{D}}$. We cannot directly apply Theorem 3.1 to conclude that ϕ and ψ are equisatisfiable since we have not assumed that $\mathfrak{D}$ is homomorphism ω-compact. The proof of the following results shows that, even without this assumption, we obtain the stronger result that ψ is a first-order abstract projective definition of ϕ.

Corollary 4.1. *Let $\mathfrak{D}$ be a unary concrete domain that is closed under negation. Then, every* $\mathsf{FOL}(\mathfrak{D})$ *sentence has an abstract projective definition in* FOL.

Proof. Let ϕ be a $\mathsf{FOL}(\mathfrak{D})$ sentence and ψ the FOL sentence obtained by the rewriting process described above. First, we show that every model of ψ is an abstract model of ϕ. Let $\mathfrak{I}$ be a model of ψ. Since $\mathfrak{D}$ is unary, the constraint system $\Gamma_{\mathfrak{I}}$ considered in the proof of Theorem 3.1 contains all expressions $P^f(x_d)$ such that $P^f(d)$ holds in $\mathfrak{I}$ for f a feature name and $d \in I$. For every feature name f and $d \in I$, let $\Gamma_{d,f}$ be the subsystem of $\Gamma_{\mathfrak{I}}$ containing all and only expressions of the form $P^f(x_d)$. We notice that each of these subsystems is finite, and that they partition $\Gamma_{\mathfrak{I}}$. In particular, $\Gamma_{\mathfrak{I}}$ is satisfiable in $\mathfrak{D}$ iff $\Gamma_{d,f}$ is satisfiable in $\mathfrak{D}$ for every f and $d \in I$. The satisfiability of each such $\Gamma_{d,f}$ in $\mathfrak{D}$ is a consequence of the fact that $\mathfrak{I}$ is a model of ψ, and thus of $\Psi^{\mathfrak{D}}$. Otherwise, $\Psi^{\mathfrak{D}}$ would contain the sentence $\forall x.\Gamma_{d,f} \to \bot$ and this would lead to a contradiction. We conclude that $\Gamma_{\mathfrak{I}}$ has a solution h in $\mathfrak{D}$, which we use as in the proof of Theorem 3.1 to define an interpretation $\mathfrak{F}$ of feature names such that $(\mathfrak{I}, \mathfrak{F})$ is a model of ϕ.

Second, we must show that every abstract model ϕ can be extended to a model of ψ by interpreting the new predicates of the form P^f and Def_f appropriately. This can be done exactly as in the proof of Theorem 3.1. □

Recall that, in the proof of Theorem 3.1, we used the downward Löwenheim-Skolem property of first-order logic to ensure that the constraint system $\Gamma_{\mathfrak{I}}$ is countable, a necessary requirement to be able to apply homomorphism ω-compactness. In the proof of Corollary 4.1, this was not possible since we had to show that the given model of ψ is an abstract model of ϕ. Fortunately, the fact that we can reduce satisfiability of $\Gamma_{\mathfrak{I}}$ to that of the finite systems $\Gamma_{d,f}$ allowed us to dispense with this step and the requirement that $\mathfrak{D}$ is homomorphism ω-compact.

For $\mathcal{ALC}(\mathfrak{D})$ TBoxes $\mathcal{T}$ we can strengthen Corollary 4.1 by introducing a TBox $\mathcal{T}^{\mathfrak{D}}$ that takes on the role of $\Psi^{\mathfrak{D}}$ in the $\mathsf{FOL}(\mathfrak{D})$ setting. First, we introduce fresh concept names P^f and Def_f for every feature name f and unary predicate P of $\mathfrak{D}$ that occur in a concrete domain restriction of $\mathcal{T}$. We denote with $\mathcal{T}^{\mathsf{FOL}}$ the $\mathcal{ALC}$ TBox from $\mathcal{T}$ obtained by replacing $\exists f.P(x)$ with P_f, $\exists rf.P(x)$ with $\exists r.P_f$, $\forall f.P(x)$ with $\neg\,\mathrm{Def}_f \sqcup P^f$ and $\forall rf.P(x)$ with $\forall r.(\neg\,\mathrm{Def}_f \sqcup P^f)$. The $\mathcal{ALC}$ TBox $\mathcal{T}^{\mathfrak{D}}$ consists of the following CIs:

- $P^f \sqsubseteq \mathrm{Def}_f$ and $\neg P^f \sqsubseteq P^f_c \sqcup \neg\,\mathrm{Def}_f$ for every feature name f and unary relation P over $\mathfrak{D}$ occurring in $\mathcal{T}$,
- $\bigsqcap \Gamma \sqsubseteq \bot$ for every feature name f and every finite set Γ of concept names P^f s.t. the constraint system $\{P(x) \mid P^f \in \Gamma\}$ is unsatisfiable in $\mathfrak{D}$.

Then, $\mathcal{T}' := \mathcal{T}^{\mathsf{FOL}} \cup \mathcal{T}^{\mathfrak{D}}$ acts as the sentence ψ did in the proof of Corollary 4.1.

Corollary 4.2. *Let $\mathfrak{D}$ be a unary concrete domain that is closed under negation. Then, every $\mathcal{ALC}(\mathfrak{D})$ TBox has an abstract projective definition in $\mathcal{ALC}$.*

Decidability results. Note that, in the setting introduced in this subsection, the $\mathsf{FOL}(\mathfrak{D})$ sentence $\Psi^{\mathfrak{D}}$ belongs both to the guarded and the two-variable fragment with counting of first-order logic, which are known to be decidable [2, 15, 22, 23]. Therefore, if the sentence ϕ falls into one of these fragments, defined analogously to their first-order counterparts, it follows that the abstract projective definition ψ of ϕ used in Corollary 4.1 also falls into this fragment. To ensure that satisfiability of $\mathsf{FOL}(\mathfrak{D})$ sentences falling into one of these fragments is decidable, it is necessary to guarantee that $\Psi^{\mathfrak{D}}$ can effectively be computed. This is the case if checking satisfiability of a finite constraint system for $\mathfrak{D}$ is decidable.

Corollary 4.3. *Let $\mathfrak{D}$ be a unary concrete domain that is closed under negation. If constraint satisfiability for $\mathfrak{D}$ is decidable, then satisfiability of sentences in the guarded or the two-variable fragment with counting of* $\mathsf{FOL}(\mathfrak{D})$ *is decidable.*

The first-order translations of many DLs considered in the literature actually belong to the guarded or the two-variable fragment with counting. Since, in the unary case, the translations of concrete domain restrictions into $\mathsf{FOL}(\mathfrak{D})$ given in (2) also belong to these fragments, the above corollary yields decidability results for a great number of DLs extended with unary and decidable concrete domains. Note, however, that this does not cover the decidability result for $\mathcal{SHOQ}$ extended with unary concrete domains in [16] since the transitivity of roles specifiable in that DL cannot be expressed in the guarded or the two-variable fragment with counting.

4.2 In(equality) causes non-definability

We have seen above that the restriction to a unary concrete domain $\mathfrak{D}$ ensures that every $\mathtt{FOL}(\mathfrak{D})$ sentence has an abstract projective definition in FOL. This also implies that $\mathtt{FOL}(\mathfrak{D})$ satisfies the upward Löwenheim-Skolem property. Without the restriction to predicates of arity 1, this need no longer be the case. In fact, Example 2.4 demonstrates that there exists a concrete domain $\mathfrak{D}$ and an $\mathcal{ALC}(\mathfrak{D})$ TBox $\mathcal{T}$ such that $\mathcal{T}$ does not have an abstract projective definition in FOL. In addition, the proof of Corollary 3.2 shows that, for the concrete domain $\mathfrak{Q}_=$, the logic $\mathtt{FOL}(\mathfrak{Q}_=)$ does not satisfy the upward Löwenheim-Skolem property. In the following, we extend these negative results from single examples to a large class of concrete domains.

Analyzing the two concrete examples, we see that they crucially depend on the fact that (in)equality can be expressed in the concrete domain under consideration. Following [10], we say that $\mathfrak{D}$ is *jointly diagonal (JD)* if equality between elements of D can be expressed using a quantifier-free formula $\psi_=(x,y)$ over the predicates contained in the signature of $\mathfrak{D}$.[2] In [10], JD is part of the definition of ω-admissibility, and thus all ω-admissible concrete domains exhibited there satisfy this property.

Theorem 4.4. *Let $\mathfrak{D}$ be a jointly diagonal, at most countable concrete domain. Then,* $\mathtt{FOL}(\mathfrak{D})$ *does not have the upward Löwenheim-Skolem property.*

Proof. Assume that $\psi_=(x,y)$ is the quantifier-free formula that expresses equality between elements of D. Let $\psi_=^f(x,y)$ be the $\mathtt{FOL}(\mathfrak{D})$ formula obtained by replacing every atom $P(x_1,\dots,x_n)$ in $\psi_=$ with $P(f,\dots,f)(x_1,\dots,x_n)$. Similarly to the proof of Corollary 3.2, we define the $\mathtt{FOL}(\mathfrak{D})$ sentence

$$\phi_{\mathrm{up}} := \forall x,y.\,\mathrm{Def}(f)(x) \land (x \neq y \to \neg\psi_=^f(x,y)),$$

which enforces that the interpretation of f is a total and injective function from the domain of any abstract model of ϕ_{up} into D. Thus, no abstract model of ϕ_{up} can have an uncountable domain. □

In Corollary 3.3 we use closure under disjoint union of models of $\mathcal{ALC}(\mathfrak{D})$ TBoxes to show that $\mathcal{ALC}(\mathfrak{D})$ has the upward Löwenheim-Skolem property. However, the fact that such a TBox then always has an uncountable model is not sufficient to apply the argument used in Example 2.4 to show that there exists an $\mathcal{ALC}(\mathfrak{D})$ TBoxes that have no abstract projective first-order definition. In fact, such an uncountable model could be the uncountable disjoint union of countable models, and injectivity of the feature name f can possible only be enforced on the countable sub-models. This is why we needed the formula τ in the proof given in that example, which states that any two distinct elements of the interpretation domain are linked by the role r. To adapt the idea underlying this proof to our more general setting, we make an additional assumption on the formula $\psi_=(x,y)$ defining equality.

Theorem 4.5. *Let $\mathfrak{D}$ be an at most countable concrete domain that is closed under negation and is JD, and assume that there is a quantifier-free definition of equality over $\mathfrak{D}$ that uses only binary relations. Then, there is an $\mathcal{ALC}(\mathfrak{D})$ TBox that has no abstract projective definition in first-order logic.*

Proof. Let $\psi_=(x,y)$ be a quantifier-free definition of equality over $\mathfrak{D}$ that uses only binary relations. Let P be a binary relation that occurs in $\psi_=(x,y)$ and P_c its complement, whose existence is guaranteed by the assumption of closure under negation. We can force a feature name f to act as a total function (in the spirit of the CI $\top \sqsubseteq \exists f,f.{=}(x_1,x_2)$ used in Example 2.4) with the $\mathcal{ALC}(\mathfrak{D})$ TBox

$$\mathcal{T}_{\mathrm{tot}} := \{\top \sqsubseteq (\exists f,f.P(x_1,x_2)) \sqcup (\exists f,f.P_c(x_1,x_2))\}.$$

Equivalence to requiring totality of f follows from the fact that, for every $d \in D$, the concrete domain $\mathfrak{D}$ satisfies either $P(d,d)$ or $P_c(d,d)$.

We combine our assumptions about $\psi_=(x,y)$ and closure under negation to obtain a quantifier-free and *positive* formula $\psi_{\neq}(x,y)$ that defines inequality over $\mathfrak{D}$ and uses only binary relations of $\mathfrak{D}$. To ensure that $\psi_{\neq}(x,y)$ does not contain negated predicates, we take the negation-normal form of $\neg\psi_=(x,y)$ and replace every negated occurrence of P with its complement P_c. We introduce for every binary relation P that occurs in $\psi_{\neq}(x,y)$ a fresh role name r_P and a CI $\top \sqsubseteq \forall f, r_P f.P(x_1,x_2)$ and call $\mathcal{T}_{\neq}$ the resulting TBox.

Let $\mathcal{T} := \mathcal{T}_{\mathrm{tot}} \cup \mathcal{T}_{\neq}$ and assume, by contradiction, that $\mathcal{T}$ is abstractly projectively equivalent to a first-order sentence ϕ. The interpretation $\mathfrak{I}$ with countable domain $I := D$ and $r_P^{\mathfrak{I}} := P^D$ is an abstract model of $\mathcal{T}$, where we interpret the feature name f using the identity over D. Then, $\mathfrak{I}$ can be extended to a model $\mathfrak{I}'$ of ϕ. Using the upward Löwenheim-Skolem property of first-order logic, we find an uncountable interpretation $\mathfrak{J}$ that is elementary equivalent to $\mathfrak{I}'$ in first-order logic (apply the property to the first-order theory of $\mathfrak{I}'$, which is trivially satisfied by $\mathfrak{I}'$). This implies that $\mathfrak{J}$ satisfies ϕ and thus, by assumption, we can find an interpretation $f^{\mathfrak{F}}$ of f such that $(\mathfrak{J},\mathfrak{F})$ is a model of $\mathcal{T}$.

Let d, e be two distinct elements of J. Assuming that $\psi_{\neq}^r(x,y)$ is the formula obtained by replacing every occurrence of $P(x,y)$ in $\psi_{\neq}(x,y)$ with $r_P(x,y)$, we observe that both $\mathfrak{I}$ and $\mathfrak{I}'$ satisfy the first-order sentence

$$\forall x,y.(x \neq y) \leftrightarrow \psi_{\neq}^r(x,y).$$

Since $\mathfrak{J}$ is elementary equivalent to $\mathfrak{I}'$, it also satisfies the above sentence and thus $(d,e) \in (\psi_{\neq}^r)^{\mathfrak{J}}$.

Since $(\mathfrak{J},\mathfrak{F})$ is a model of $\mathcal{T}_{\mathrm{tot}}$, both $f^{\mathfrak{F}}(d)$ and $f^{\mathfrak{F}}(e)$ must be defined. The fact that $(\mathfrak{J},\mathfrak{F})$ is a model of $\mathcal{T}_{\neq}$ ensures that $(d,e) \in (r_P)^{\mathfrak{J}}$ implies $(f^{\mathfrak{F}}(d), f^{\mathfrak{F}}(e)) \in P^D$ for every predicate P occurring in $\psi_{\neq}(x,y)$. Therefore, $(f^{\mathfrak{F}}(d), f^{\mathfrak{F}}(e)) \in \psi_{\neq}^D$ holds, and consequently $f^{\mathfrak{F}}(d) \neq f^{\mathfrak{F}}(e)$, which implies that $f^{\mathfrak{F}}$ is an injective function. This leads to a contradiction since we know that the domain D of $\mathfrak{D}$ is at most countable, but the domain J of $\mathfrak{J}$ is uncountable, and $f^{\mathfrak{F}}$ is supposed to be an injective function from J to D. We conclude that $\mathcal{T}$ is not abstractly projectively equivalent to any first-order sentence. □

To conclude this section, let us point out that the assumptions made in this theorem are not very restrictive. As already mentioned above, JD is part of the definition of ω-admissibility in [10].

[2] Note that the equality predicate cannot be employed in such a formula unless it is explicitly contained in the signature of $\mathfrak{D}$.

Whereas [10] does not assume closure under negation of the set of concrete domain predicates, it requires that ω-admissible concrete domains are jointly exhaustive and pairwise disjoint (JEPD). It is easy to see that JEPD can replace closure under negation in the proof of the above theorem since the complement of any relation of $\mathfrak{D}$ can then be expressed as the union of the other relations. Finally, note that the original work introducing ω-admissibility [20] assumed that all relations are binary, and many of the ω-admissible concrete domains exhibited in [10] also satisfy this restriction.

5 CONCLUSION

We have introduced the notion of abstract expressive power of a logic (FOL or a DL) with concrete domain, which is determined by which classes of abstract models (where one abstracts away the interpretation of feature names) can be defined by sentences of this logic. Our first main result is that such classes of abstract models share compactness and the downward Löwenheim-Skolem property with the ones definable by FOL if the employed concrete domain satisfies some reasonable model-theoretic assumptions. Under an additional computability assumption, the construction used to show these results also provides us with an effective procedure for enumerating all unsatisfiable sentences of the logic with concrete domain. An interesting topic for future research is to check which other properties of FOL (e.g., Craig interpolation [21] or the 0-1 law [13]) hold for (fragments of) $\mathsf{FOL}(\mathfrak{D})$, depending on certain properties satisfied by $\mathfrak{D}$. It is well-known that $\mathcal{ALC}$ is the fragment of FOL that is closed under bisimulation [11]. It would be interesting to see whether a similar result holds for $\mathcal{ALC}(\mathfrak{D})$ and $\mathsf{FOL}(\mathfrak{D})$, based on an appropriate notion of bisimulation.

Our second main result is that, although sharing interesting properties with FOL, sentences of logics with concrete domain are often not (projectively) definable in FOL. An exception are logics with unary concrete domains, where we obtain FOL definability under weak additional assumptions. Given a logic with concrete domain, inexpressibility in FOL does not mean that none of its sentences are definable in FOL. Thus, one can ask if the existence of a (projective) definition in FOL for a given sentence is decidable.

ACKNOWLEDGMENTS

This work was partially supported by the DFG TRR 248 (CPEC, grant 389792660), 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.

REFERENCES

[1] James F. Allen. 1983. Maintaining Knowledge about Temporal Intervals. *Commun. ACM* 26, 11 (1983), 832–843. https://doi.org/10.1145/182.358434

[2] Hajnal Andréka, István Németi, and Johan van Benthem. 1998. Modal Languages and Bounded Fragments of Predicate Logic. *J. Philos. Log.* 27, 3 (1998), 217–274. https://doi.org/10.1023/A:1004275029985

[3] Alessandro Artale, Vladislav Ryzhikov, and Roman Kontchakov. 2012. DL-Lite with Attributes and Datatypes. In *ECAI 2012 – 20th European Conference on Artificial Intelligence*, Luc De Raedt, Christian Bessiere, Didier Dubois, Patrick Doherty, Paolo Frasconi, Fredrik Heintz, and Peter J. F. Lucas (Eds.). IOS Press, 61–66. https://doi.org/10.3233/978-1-61499-098-7-61

[4] Franz Baader. 1996. A Formal Definition for the Expressive Power of Terminological Knowledge Representation Languages. *J. of Logic and Computation* 6 (1996), 33–54. https://doi.org/10.1093/logcom/6.1.33

[5] Franz Baader and Filippo De Bortoli. 2019. On the Expressive Power of Description Logics with Cardinality Constraints on Finite and Infinite Sets. In *Proc. of the 12th Int. Symposium on Frontiers of Combining Systems (FroCoS'19) (Lecture Notes in Computer Science, Vol. 11715)*, Andreas Herzig and Andrei Popescu (Eds.). Springer-Verlag, 203–219. https://doi.org/10.1007/978-3-030-29007-8_12

[6] Franz Baader and Philipp Hanschke. 1991. A Scheme for Integrating Concrete Domains into Concept Languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991*, John Mylopoulos and Raymond Reiter (Eds.). Morgan Kaufmann, 452–457. http://ijcai.org/Proceedings/91-1/Papers/070.pdf

[7] Franz Baader and Philipp Hanschke. 1992. Extensions of Concept Languages for a Mechanical Engineering Application. In *GWAI-92: Advances in Artificial Intelligence, 16th German Conference on Artificial Intelligence (Lecture Notes in Computer Science, Vol. 671)*, Hans Jürgen Ohlbach (Ed.). Springer, 132–143. https://doi.org/10.1007/BFb0018999

[8] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. 2017. *An Introduction to Description Logic.* Cambridge University Press. https://doi.org/10.1017/9781139025355

[9] Franz Baader and Jakub Rydval. 2020. Description Logics with Concrete Domains and General Concept Inclusions Revisited. In *Automated Reasoning (Lecture Notes in Computer Science)*, Nicolas Peltier and Viorica Sofronie-Stokkermans (Eds.). Springer International Publishing, Cham, 413–431. https://doi.org/10.1007/978-3-030-51074-9_24

[10] Franz Baader and Jakub Rydval. 2022. Using Model Theory to Find Decidable and Tractable Description Logics with Concrete Domains. *Journal of Automated Reasoning* 66, 3 (Aug. 2022), 357–407. https://doi.org/10.1007/s10817-022-09626-2

[11] Patrick Blackburn, Maarten de Rijke, and Yde Venema. 2001. *Modal Logic.* Cambridge Tracts in Theoretical Computer Science, Vol. 53. Cambridge University Press. https://doi.org/10.1017/CBO9781107050884

[12] Heinz-Dieter Ebbinghaus, Jörg Flum, and Wolfgang Thomas. 1994. *Mathematical Logic* (second ed.). Springer, New York, NY. https://doi.org/10.1007/978-1-4757-2355-7

[13] Ronald Fagin. 1976. Probabilities on Finite Models. *Journal of Symbolic Logic* 41, 1 (1976), 50–58. https://doi.org/10.1017/S0022481200051756

[14] Melvin Fitting. 1996. *First-Order Logic and Automated Theorem Proving* (2. ed ed.). Springer Science+Business Media, New York. https://doi.org/10.1007/978-1-4612-2360-3

[15] Erich Grädel, Martin Otto, and Eric Rosen. 1997. Two-Variable Logic with Counting is Decidable. In *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science, Warsaw, Poland, June 29 - July 2, 1997.* IEEE Computer Society, 306–317. https://doi.org/10.1109/LICS.1997.614957

[16] Ian Horrocks and Ulrike Sattler. 2001. Ontology Reasoning in the SHOQ(D) Description Logic. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001*, Bernhard Nebel (Ed.). Morgan Kaufmann, 199–204.

[17] Natasha Kurtonina and Maarten de Rijke. 1999. Expressiveness of Concept Expressions in First-Order Description Logics. *Artificial Intelligence* 107, 2 (1999), 303–333. https://doi.org/10.1016/S0004-3702(98)00109-X

[18] Carsten Lutz. 2002. Description Logics with Concrete Domains—A Survey. In *Advances in Modal Logic 4*, Philippe Balbiani, Nobu-Yuki Suzuki, Frank Wolter, and Michael Zakharyaschev (Eds.). King's College Publications, 265–296.

[19] Carsten Lutz. 2004. NEXP TIME-complete description logics with concrete domains. *ACM Transactions on Computational Logic (TOCL)* 5, 4 (2004), 669–705. https://doi.org/10.1145/1024922.1024925

[20] Carsten Lutz and Maja Miličić. 2007. A Tableau Algorithm for Description Logics with Concrete Domains and General TBoxes. *Journal of Automated Reasoning* 38, 1 (April 2007), 227–259. https://doi.org/10.1007/s10817-006-9049-7

[21] Roger Lyndon. 1959. An Interpolation Theorem in the Predicate Calculus. *Pacific J. Math.* 9, 1 (March 1959), 129–142. https://msp.org/pjm/1959/9-1/p12.xhtml

[22] Leszek Pacholski, Wieslaw Szwast, and Lidia Tendera. 1997. Complexity of Two-Variable Logic with Counting. In *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science.* IEEE Computer Society, 318–327. https://doi.org/10.1109/LICS.1997.614958

[23] Ian Pratt-Hartmann. 2005. Complexity of the Two-Variable Fragment with Counting Quantifiers. *J. Log. Lang. Inf.* 14, 3 (2005), 369–395. https://doi.org/10.1007/s10849-005-5791-1

[24] David A. Randell, Zhan Cui, and Anthony G. Cohn. 1992. A Spatial Logic based on Regions and Connection. In *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, Bernhard Nebel, Charles Rich, and William R. Swartout (Eds.). Morgan Kaufmann, 165–176.

[25] Ognjen Savkovic and Diego Calvanese. 2012. Introducing Datatypes in DL-Lite. In *ECAI 2012 – 20th European Conference on Artificial Intelligence*, Luc De Raedt, Christian Bessiere, Didier Dubois, Patrick Doherty, Paolo Frasconi, Fredrik Heintz, and Peter J. F. Lucas (Eds.). IOS Press, 720–725. https://doi.org/10.3233/978-1-61499-098-7-720

[26] Manfred Schmidt-Schauß and Gert Smolka. 1991. Attributive Concept Descriptions with Complements. *Artif. Intell.* 48, 1 (1991), 1–26. https://doi.org/10.1016/0004-3702(91)90078-X