Reasoning in OWL 2 EL with Hierarchical Concrete Domains

Francesco Kriegel[1,2]

[1]Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany
[2]Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI)
francesco.kriegel@tu-dresden.de

Abstract. The $\mathcal{EL}$ family of description logics facilitates efficient polynomial-time reasoning and has been standardized as the profile OWL 2 EL of the Web Ontology Language. $\mathcal{EL}$ can represent and reason not only with symbolic knowledge but also with concrete knowledge expressed by numbers, strings, and other concrete datatypes. Such concrete domains must be convex to avoid introducing disjunctions "through the backdoor." However, existing concrete domains provide only limited utility. In order to overcome this issue, we introduce a novel form of concrete domains based on semi-lattices. They are convex by design and can thus be integrated into Horn-DLs such as $\mathcal{EL}$. Moreover, they allow for FBoxes to express dependencies between concrete features. We describe four instantiations concerned with real intervals, 2D-polygons, regular languages, and graphs.

1 Introduction

Concrete domains can be integrated in description logics (DLs) in order to refer to concrete knowledge expressed by numbers, strings, and other concrete datatypes [7]. They have mainly been investigated with DLs that are not Horn, such as $\mathcal{ALC}$ and its extensions, regarding decidability and complexity [14, 18, 20, 49, 50, 51], reasoning procedures [26, 27, 50, 51, 52, 58], an algebraic characterization [12, 59], and their expressive power [3, 6].

For computationally tractable description logics, other conditions on the concrete domains than above must be imposed. Suitable for the $\mathcal{EL}$ family are p-admissible concrete domains [4]: through them it is not possible to introduce disjunction into the logical domain so that the DL part retains its Horn character and, moreover, they guarantee that reasoning involving both the logical and the concrete domain remains tractable. Concrete domains have also been integrated with DL-Lite [2].

Existing p-admissible concrete domains for $\mathcal{EL}$ provide only limited utility. Using the concrete domain $\mathcal{D}_{\mathbb{Q},\mathsf{diff}}$ [4], we could express with the concept inclusions $(\mathsf{sys} \geq 140) \sqsubseteq \mathsf{Hypertension}$ and $(\mathsf{dia} \geq 90) \sqsubseteq \mathsf{Hypertension}$ that a systolic blood pressure of 140 or higher indicates hypertension, as does a diastolic blood pressure of at least 90. Since the opposite relations $\leq$ are not available to ensure convexity, neither non-elevated blood pressure (dia. $<$ 120 and sys. $<$ 70) nor elevated blood pressure (dia. between 120 and 140, and sys. between 70 and 90) are expressible. Mixed inequalities $<$, $\leq$, $>$, and $\geq$ may be used under certain limitations which of them may occur in left-hand sides and, respectively, in right-hand sides of concept inclusions [53]. While this retains convexity of the concrete domain, reasoning is then rather impaired since the usual completion procedure is only complete for consistency and classification, but not for subsumption.

An algebraic characterization of p-admissible concrete domains has put forth a further concrete domain $\mathcal{D}_{\mathbb{Q},\mathsf{lin}}$, which supports linear combinations of numerical features [11, 13]. For instance, the concept inclusion $\top \sqsubseteq (\mathsf{sys} - \mathsf{dia} - \mathsf{pp} = 0)$, where $-$ is the difference operation in real arithmetic, expresses that the pulse pressure is the difference between the systolic and the diastolic blood pressure. In the medical domain, the combined expressivity of $\mathcal{D}_{\mathbb{Q},\mathsf{diff}}$ and $\mathcal{D}_{\mathbb{Q},\mathsf{lin}}$ would be useful since then with the concept inclusion $\mathsf{ICUPatient} \sqcap (\mathsf{pp} > 50) \sqsubseteq \mathsf{NeedsAttention}$ it could be expressed that intensive-care patients with a pulse pressure exceeding 50 need attention — but this combination is not convex anymore [1].

We introduce a novel form of concrete domains based on semi-lattices. A semi-lattice $(L, \leq, \wedge)$ consists of a set L, a partial order $\leq$, and a binary meet operation $\wedge$. The elements of L are taken as concrete values, and $\leq$ is understood as an "information order," i.e. $p \leq q$ means that p is more specific than q, like a subsumption order between concepts. The meet operation $\wedge$ is used to combine two values p and q to their meet value $p \wedge q$, which is the most general value that is more specific than both p and q. For instance, real intervals form a semi-lattice with subset inclusion $\subseteq$ as partial order and intersection $\cap$ as meet operation. With that, the statement $\mathsf{NonElevatedBP} \equiv (\mathsf{sys} \subseteq [0, 120)) \sqcap (\mathsf{dia} \subseteq [0, 70))$ defines non-elevated blood pressure, where $[0, 120)$ and $[0, 70)$ are real intervals.

Our new *hierarchical concrete domains* are convex by design, simply because a general value of a feature (such as $\mathsf{sys} \subseteq [0, 120)$) does not imply the disjunction of all more specific feature values (such as $\mathsf{sys} \subseteq [0, 0]$, $\mathsf{sys} \subseteq [1, 1]$, $\ldots$, $\mathsf{sys} \subseteq [119, 119]$). Atomic feature values are supported nonetheless when these are available as atoms in the semi-lattice. For instance, a specific numerical value p is represented by the singleton interval $[p, p]$ (which equals the one-element set $\{p\}$).

In addition, we introduce *FBoxes* consisting of *feature inclusions* that describe dependencies between features as well as aggregations of features. For instance, through the feature inclusion $\mathsf{pp} \subseteq \mathsf{sys} - \mathsf{dia}$, where $-$ is the difference operation in real interval arithmetic, we can obtain an interval value of the pulse pressure given intervals of the systolic and the diastolic blood pressure. With the concept inclusion $\mathsf{ICUPatient} \sqcap (\mathsf{pp} \subseteq (50, \infty)) \sqsubseteq \mathsf{NeedsAttention}$ we can now express that intensive-care patients having a pulse pressure above 50 need attention and, unlike in the combination of $\mathcal{D}_{\mathbb{Q},\mathsf{diff}}$ and $\mathcal{D}_{\mathbb{Q},\mathsf{lin}}$, computationally reason with that in polynomial time.

We provide four instantiations of hierarchical concrete domains based on real intervals, 2D-polygons, regular languages, and graphs. The former two are not only convex, but indeed p-admissible, i.e. equipping a DL from the $\mathcal{EL}$ family with them facilitates polynomial-time reasoning. In particular, we can employ linear programming for reasoning in the interval domain when the FBox is affine.

The regular-language domain is also convex (again, by design) but requires exponential time for reasoning. However, this only affects the concrete-domain reasoning itself so that reasoning in the logical $\mathcal{EL}$ part still runs in polynomial time. This holds similarly for the graph domain.

Of practical relevance is that our hierarchical concrete domains can be seamlessly integrated into the completion procedure and the ELK reasoner [4, 5, 40, 42]. We demonstrate this for the case where nominals must be used safely, i.e. nominals must not occur in conjunctions and right-hand sides of concept inclusions must not be single nominals. We conjecture that full support for nominals can be achieved in the same way as without concrete domains [41].

Proofs and more technical details can be found in the extended version [46].

2 Preliminaries

We work with the description logic $\mathcal{EL}^{++}[\mathcal{D}]$ (OWL 2 EL) where $\mathcal{D}$ is a P-admissible concrete domain (as defined below). Consider a set $\mathbf{C}$ of *atomic concepts*, a set $\mathbf{R}$ of *roles*, a set $\mathbf{I}$ of *individuals*, a set $\mathbf{F}$ of *features*, and a set $\mathbf{P}$ of *predicates* where each $P \in \mathbf{P}$ has an arity $\mathsf{ar}(P) \in \mathbb{N}$. There are two special concepts $\bot$ and $\top$ with fixed meaning. A *constraint* has the form $\exists f_1, \ldots, f_k.P$ where P is a k-ary predicate and $f_1, \ldots, f_k$ are features. *Compound concepts* are built by

$$C ::= \bot \mid \top \mid \{i\} \mid A \mid \exists f_1, \ldots, f_k.P \mid C \sqcap C \mid \exists r.C$$

where A ranges over all atomic concepts, r over all roles, i over all individuals, and $\exists f_1, \ldots, f_k.P$ over all constraints. A *knowledge base (KB)* is a finite set of *concept inclusions (CIs)* $C \sqsubseteq D$ concerning concepts C and D, *role inclusions (RIs)* $R \sqsubseteq s$ involving *role chains* generated by $R ::= \varepsilon \mid R_1$, $R_1 ::= r \mid R_1 \circ R_1$ and roles s, and *range inclusions* $\mathsf{Ran}(r) \sqsubseteq C$ referring to roles r and concepts C — but every $\mathcal{EL}^{++}[\mathcal{D}]$ KB must satisfy an additional condition as explained in Section 4.

As syntactic sugar, we have *concept assertions* $\{i\} \sqsubseteq C$ (also written $i : C$), *role assertions* $\{i\} \sqsubseteq \exists r.\{j\}$ (also written $(i, j) : r$), *domain inclusions* $\exists r.\top \sqsubseteq C$ (also written $\mathsf{Dom}(r) \sqsubseteq C$), and *role exclusions* $\exists r_1. \ldots \exists r_n.\top \sqsubseteq \bot$ (also written $r_1 \circ \cdots \circ r_n \sqsubseteq \bot$). Statements $C \sqsubseteq \bot$ are also called *concept exclusions*, and $C \equiv D$ is a *concept equivalence* that stands for the two CIs $C \sqsubseteq D$ and $D \sqsubseteq C$. Each KB $\mathcal{K}$ can be subdivided into an *ABox* $\mathcal{A}$ consisting of all concept and role assertions, an *RBox* $\mathcal{R}$ consisting of all role inclusions and exclusions, and a *TBox* $\mathcal{T}$ consisting of the remaining statements. The TBox together with the RBox is also called an *ontology* $\mathcal{O}$.

The semantics are defined through the fixed concrete domain $\mathcal{D}$ and all interpretations $\mathcal{I}$. The *concrete domain* $\mathcal{D} := (\mathsf{Dom}(\mathcal{D}), \cdot^{\mathcal{D}})$ consists of a set $\mathsf{Dom}(\mathcal{D})$ of *values* and an interpretation function $\cdot^{\mathcal{D}}$ that sends each predicate $P \in \mathbf{P}$ to a relation over $\mathsf{Dom}(\mathcal{D})$ with arity $\mathsf{ar}(P)$, i.e. $P^{\mathcal{D}} \subseteq \mathsf{Dom}(\mathcal{D})^{\mathsf{ar}(P)}$.

If the predicate P in a constraint $\exists f_1, \ldots, f_k.P$ is defined through a mathematical expression or a logical formula with k free variables, then we may represent the constraint also through this expression/formula but with the free variables replaced by the features $f_1, \dots, f_k$. For instance, the constraint $\mathsf{sys} - \mathsf{dia} - \mathsf{pp} = 0$ from the introduction represents $\exists \mathsf{sys}, \mathsf{dia}, \mathsf{pp}. P_{(1,-1,-1),0}$ where $(P_{(1,-1,-1),0})^{\mathcal{D}} := \{ (x, y, z) \mid x - y - z = 0 \}$.

An interpretation $\mathcal{I} := (\mathsf{Dom}(\mathcal{I}), \cdot^{\mathcal{I}})$ consists of a non-empty set $\mathsf{Dom}(\mathcal{I})$, called *domain*, and an interpretation function $\cdot^{\mathcal{I}}$ that maps each atomic concept $A \in \mathbf{C}$ to a subset $A^{\mathcal{I}}$ of $\mathsf{Dom}(\mathcal{I})$, each role $r \in \mathbf{R}$ to a binary relation $r^{\mathcal{I}}$ over $\mathsf{Dom}(\mathcal{I})$, each individual $i \in \mathbf{I}$ to an element $i^{\mathcal{I}}$ of $\mathsf{Dom}(\mathcal{I})$, and each feature $f \in \mathbf{F}$ to a partial function $f^{\mathcal{I}}$ from $\mathsf{Dom}(\mathcal{I})$ to $\mathsf{Dom}(\mathcal{D})$. The interpretation function $\cdot^{\mathcal{I}}$ is extended to compound concepts as follows: $\bot^{\mathcal{I}} := \emptyset$, $\top^{\mathcal{I}} := \mathsf{Dom}(\mathcal{I})$, $\{i\}^{\mathcal{I}} := \{i^{\mathcal{I}}\}$, $(\exists f_1, \dots, f_k. P)^{\mathcal{I}} := \{ x \mid x \in \mathsf{Dom}(f_1^{\mathcal{I}}) \cap \dots \cap \mathsf{Dom}(f_k^{\mathcal{I}})$ and $(f_1^{\mathcal{I}}(x), \dots, f_k^{\mathcal{I}}(x)) \in P^{\mathcal{D}} \}$, $(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$, and $(\exists r. C)^{\mathcal{I}} := \{ x \mid$ there is y s.t. $(x, y) \in r^{\mathcal{I}}$ and $y \in C^{\mathcal{I}} \}$. Role chains are interpreted by $\varepsilon^{\mathcal{I}} := \{ (x, x) \mid x \in \mathsf{Dom}(\mathcal{I}) \}$ and $(R \circ S)^{\mathcal{I}} := \{ (x, z) \mid$ there is y s.t. $(x, y) \in R^{\mathcal{I}}$ and $(y, z) \in S^{\mathcal{I}} \}$, and role ranges are interpreted as $\mathsf{Ran}(r)^{\mathcal{I}} := \{ y \mid$ there is x s.t. $(x, y) \in r^{\mathcal{I}} \}$.

$\mathcal{I}$ *satisfies* a concept/role/range inclusion $X \sqsubseteq Y$, written $\mathcal{I} \models X \sqsubseteq Y$, if $X^{\mathcal{I}} \subseteq Y^{\mathcal{I}}$. If $\mathcal{I}$ satisfies all inclusions in a KB $\mathcal{K}$, then $\mathcal{I}$ is a *model* of $\mathcal{K}$, written $\mathcal{I} \models \mathcal{K}$. If $\mathcal{K}$ has a model, then it is *consistent*, and otherwise *inconsistent*. $\mathcal{K}$ *entails* an inclusion $X \sqsubseteq Y$ if $X \sqsubseteq Y$ is satisfied by all models of $\mathcal{K}$, written $\mathcal{K} \models X \sqsubseteq Y$ or $X \sqsubseteq^{\mathcal{K}} Y$, and we then say that X is *subsumed by* Y w.r.t. $\mathcal{K}$. Furthermore, $\mathcal{K}$ *entails* a KB $\mathcal{L}$ if $\mathcal{K}$ entails all inclusions in $\mathcal{L}$, written $\mathcal{K} \models \mathcal{L}$.

A *constraint inclusion* is of the form $\bigsqcap \Gamma \sqsubseteq \bigsqcup \Delta$ where Γ and Δ are finite sets of constraints. $\mathcal{I}$ *satisfies* $\bigsqcap \Gamma \sqsubseteq \bigsqcup \Delta$, written $\mathcal{I} \models \bigsqcap \Gamma \sqsubseteq \bigsqcup \Delta$, if $\bigcap \{ \alpha^{\mathcal{I}} \mid \alpha \in \Gamma \} \subseteq \bigcup \{ \beta^{\mathcal{I}} \mid \beta \in \Delta \}$. Moreover, $\bigsqcap \Gamma \sqsubseteq \bigsqcup \Delta$ is *valid*, written $\mathcal{D} \models \bigsqcap \Gamma \sqsubseteq \bigsqcup \Delta$, if it is satisfied in all interpretations. It is easy to see that validity is independent of the concepts, roles, and individuals and that it suffices to consider only one domain element. To this end, a *valuation* is a partial function v from $\mathbf{F}$ to $\mathsf{Dom}(\mathcal{D})$, and it *satisfies* $\exists f_1, \dots, f_k. P$ if $(v(f_1), \dots, v(f_k)) \in P^{\mathcal{D}}$. Now, $\bigsqcap \Gamma \sqsubseteq \bigsqcup \Delta$ is *valid* iff., for each valuation v, if v satisfies all $\alpha \in \Gamma$, then v satisfies some $\beta \in \Delta$.

We say that $\mathcal{D}$ is P-*admissible* if satisfiability of constraint conjunctions as well as validity of constraint inclusions are decidable in polynomial time and, moreover, $\mathcal{D}$ is *convex*, i.e. for each valid constraint inclusion $\bigsqcap \Gamma \sqsubseteq \bigsqcup \Delta$, there is a constraint $\beta \in \Delta$ such that $\bigsqcap \Gamma \sqsubseteq \beta$ is valid. We can use multiple P-admissible concrete domains by forming their disjoint union, which is P-admissible too.

3 Hierarchical Concrete Domains

A *semi-lattice* $\mathbf{L} := (L, \leq, \wedge)$ consists of a set L, a partial order $\leq$ on L, and a binary meet operation $\wedge$ on L, i.e. the following hold for all $p, q, p_1, p_2, p_3 \in L$:

(SL1) $p \leq p$ for each $p \in L$ (reflexive)
(SL2) if $p \leq q$ and $q \leq p$, then $p = q$ (anti-symmetric)
(SL3) if $p_1 \leq p_2$ and $p_2 \leq p_3$, then $p_1 \leq p_3$ (transitive)
(SL4) $p_1 \wedge p_2 \leq p_1$ and $p_1 \wedge p_2 \leq p_2$
(SL5) if $q \leq p_1$ and $q \leq p_2$, then $q \leq p_1 \wedge p_2$.

The strict part $<$ is defined by $p < q$ if $p \leq q$ but $q \not\leq p$, and we then say that p is *more specific than* q. Thus $p \leq q$ iff. $p < q$ or $p = q$, in which case we say that p is *more specific than or equal to* q. And $p \wedge q$ is the *meet* of p and q. It follows from the above conditions that $\wedge$ is associative, commutative, and idempotent. The finitary meet operation $\bigwedge$ is obtained from the binary one by setting $\bigwedge\{p\} := p$, $\bigwedge\{p, q\} := p \wedge q$, and $\bigwedge\{p_1, \ldots, p_n\} := p_1 \wedge \bigwedge\{p_2, \ldots, p_n\}$ whenever $n \geq 3$.

We say that $\mathbf{L}$ is *computable* if L and $\leq$ are decidable and $\wedge$ is computable. If all this is possible in polynomial time, then $\mathbf{L}$ is *polynomial-time computable.* $\mathbf{L}$ is *bounded* if it has a greatest element $\top$, i.e. $p \leq \top$ for every $p \in L$. Then we can also define a nullary meet as $\bigwedge \emptyset := \top$. In order to express impossible combinations of values, it might be convenient to add an artificial smallest element $\bot$ to the semi-lattice, i.e. $\bot \leq p$ for each $p \in L$. We then use $\bot$ to represent contradictory or ill-defined values. More specifically, $p \wedge q = \bot$ if it is impossible to combine the values p and q.

Example 1. A semi-lattice representing grades could have the values Attended, Passed, Failed, 1, 2, 3, 4, 5, 6, 1.0, 1.3, 1.7, 2.0, and so on. Its partial order $\leq$ is defined by Passed $\leq$ Attended, Failed $\leq$ Attended, $1 \leq$ Passed, $2 \leq$ Passed, $3 \leq$ Passed, $4 \leq$ Passed, $5 \leq$ Failed, $6 \leq$ Failed, $1.0 \leq 1$, $1.3 \leq 1$, $1.7 \leq 2$, $2.0 \leq 2$, etc. Here we need to add a smallest element $\bot$ since e.g. the meet of grades 1.0 and 5.0 cannot be reasonably defined.

For every KB $\mathcal{K}$ expressed in a decidable DL, the set of all concepts ordered by subsumption $\sqsubseteq^{\mathcal{K}}$ and with conjunction $\sqcap$ as meet operation is a computable, bounded semi-lattice. For each set M, $(\wp(M), \subseteq, \cap, M)$ and $(\wp(M), \supseteq, \cup, \emptyset)$ are bounded semi-lattices (where $\wp(M)$ is the powerset of M). They are only computable if restricted to finite or finitely representable subsets of M. In the following subsections we will introduce four application-relevant semi-lattices based on intervals, polygons, regular languages, and graphs.

Definition 2. *Given a bounded semi-lattice* $\mathbf{L} := (L, \leq, \wedge, \top)$*, the* hierarchical concrete domain $\mathcal{D}_{\mathbf{L}}$ *has values in* $\mathsf{Dom}(\mathcal{D}_{\mathbf{L}}) := L$ *and supports only constraints of the form* $\exists f. P_{\leq p}$*, written as* $f \leq p$*, involving a feature* f *and a value* p*. The semantics are* $(P_{\leq p})^{\mathcal{D}_{\mathbf{L}}} := \{\, q \mid q \in L \text{ and } q \leq p \,\}$ *and thus* $(f \leq p)^{\mathcal{I}} = \{\, x \mid f^{\mathcal{I}}(x) \leq p \,\}$*. Recall: this means that* f*'s value is* p *or more specific, not smaller. We assume that* $\top$ *stands for an undefined value and thus all valuations are total, i.e.* $v(f) = \top$ *means that* f *has no value under* v*. In order to represent a most general value,* $\mathbf{L}$ *contains a second-largest element* $\Box$*, i.e.* $\Box < \top$ *and* $p \leq \Box$ *for each* $p \in L \setminus \{\top\}$*. Since* $\bot$ *represents contradictory, ill-defined values, no valuation* v *assigns* $\bot$ *to any feature* f*, i.e.* $v(f) \neq \bot$*.*

Definition 3. *A* feature inclusion (FI) $f \leq H(g_1, \ldots, g_n)$ *consists of features* $f, g_1, \ldots, g_n$ *and a computable* n*-ary operation* $H\colon L^n \to L$ *that is* monotonic *in the sense that* $H(p_1, \ldots, p_n) \leq H(q_1, \ldots, q_n)$ *whenever* $p_1 \leq q_1, \ldots,$ *and* $p_n \leq q_n$ *(i.e. applying* H *to more specific values yields more specific values). A valuation* v satisfies *this FI if* $v(f) \leq H(v(g_1), \ldots, v(g_n))$*, denoted as* $v \models f \leq H(g_1, \ldots, g_n)$*. An* FBox $\mathcal{F}$ *is a finite set of FIs, and a valuation* v satisfies $\mathcal{F}$*, written* $v \models \mathcal{F}$*, if*

v satisfies every FI in $\mathcal{F}$. We call $\mathcal{F}$ acylic *if the graph $(\mathbf{F}, \{\, (f, g_1), \ldots, (f, g_n) \mid f \leq H(g_1, \ldots, g_n) \in \mathcal{F} \,\})$ is, and* cyclic *otherwise.*

The following example illustrates that FIs are "directed specifications" in the sense that values of the right-hand side features $g_1, \ldots, g_n$ yield, through the operation H, an upper bound for the value of the left-hand side feature f. However, this does not work in the other direction unless specified by other FIs.

Example 4. We use three features with interval values over the non-negative integers: sys for the systolic and dia for the diastolic blood pressure, and pp for the pulse pressure, which is the difference between the systolic and the diastolic pressure. The FI $\mathsf{pp} \subseteq \mathsf{sys} - \mathsf{dia}$ allows us to infer a value for pp when values for both sys and dia are given. The monotonic operator H in the right-hand side is $H([p_1, q_1], [p_2, q_2]) \coloneqq [p_1, q_1] - [p_2, q_2]$, and the latter value is the difference in interval arithmetic ($= [p_1 - q_2, q_1 - p_2]$ but with negative subtraction results replaced by 0). According to the semantics, an interval value of the feature pp must be a subset of $\mathsf{sys} - \mathsf{dia}$, i.e. if the latter two features are defined for an object x in a model $\mathcal{I}$ of the FI, then also $\mathsf{pp}^{\mathcal{I}}(x)$ is defined and is equal to or more specific than $H(\mathsf{sys}^{\mathcal{I}}(x), \mathsf{dia}^{\mathcal{I}}(x))$.

For instance, under the above FI the constraint inclusion $(\mathsf{sys} \subseteq [110, 120]) \sqcap (\mathsf{dia} \subseteq [60, 70]) \sqsubseteq (\mathsf{pp} \subseteq [40, 60])$ is valid since $H([110, 120], [60, 70]) = [40, 60] \subseteq [40, 60]$. Without syntactic sugar, the first constraint is $\exists \mathsf{sys}.\, P_{\subseteq[110,120]}$ involving the predicate $P_{\subseteq[110,120]} \coloneqq \{\, [p, q], (p, q), [p, q), (p, q] \mid 110 \leq p \leq q \leq 120 \,\}$.

In contrast, the constraint inclusion $(\mathsf{sys} \subseteq [110, 120]) \sqcap (\mathsf{pp} \subseteq [40, 60]) \sqsubseteq (\mathsf{dia} \subseteq [60, 70])$ is not valid w.r.t. the above FI. A countervaluation is v with $v(\mathsf{sys}) = [110, 120]$, $v(\mathsf{dia}) = [0, \infty)$, $v(\mathsf{pp}) = [40, 60]$. This is because $[110, 120] - [0, \infty) = [0, 120]$ and $[40, 60] \subseteq [0, 120]$, i.e. v satisfies the FI, but v does not satisfy the latter constraint inclusion.

Definition 5. *The semantics of the concrete domain $\mathcal{D}_{\mathbf{L}}$ can be restricted w.r.t. an FBox $\mathcal{F}$ by considering only valuations satisfying $\mathcal{F}$. That is, a constraint inclusion $\bigsqcap \Gamma \sqsubseteq \bigsqcup \Delta$ is* valid *in $\mathcal{D}_{\mathbf{L}}$ w.r.t. $\mathcal{F}$, written $\mathcal{D}_{\mathbf{L}}, \mathcal{F} \models \bigsqcap \Gamma \sqsubseteq \bigsqcup \Delta$, if this inclusion is satisfied in all valuations that satisfy $\mathcal{F}$. Whenever we write "w.r.t. $\mathcal{F}$" in the following, only valuations satisfying $\mathcal{F}$ are considered.*

Using this semantics restricted by an FBox, convexity and P-admissibility are defined as before but the latter additionally takes the FBox $\mathcal{F}$ as part of the input. The underlying semi-lattice $\mathbf{L}$ is taken into account through the computational complexity of its value set L, its partial order $\leq$, and its meet operation $\wedge$.

Definition 6. *$\mathcal{D}_{\mathbf{L}}$ is* admissible *w.r.t. $\mathcal{F}$ if $\mathcal{D}_{\mathbf{L}}$ is convex and satisfiability of constraint conjunctions as well as validity of constraint inclusions are decidable, all w.r.t. $\mathcal{F}$. For a complexity class C, we say that $\mathcal{D}_{\mathbf{L}}$ is* C-admissible *w.r.t. $\mathcal{F}$ if, all w.r.t. $\mathcal{F}$, $\mathcal{D}_{\mathbf{L}}$ is convex and satisfiability of constraint conjunctions as well as validity of constraint inclusions are in C when $\mathcal{F}$ is part of the input.*

Next, we show that a hierarchical concrete domain $\mathcal{D}_{\mathbf{L}}$ is convex w.r.t. $\mathcal{F}$ if the semi-lattice $\mathbf{L}$ is complete or well-founded or the FBox $\mathcal{F}$ is acyclic. There might be further sufficient conditions for convexity; we leave this for future research.

Definition 7. *Let* $\mathbf{L}$ *be a bounded semi-lattice and* $\mathcal{F}$ *be an FBox. Given a finite set* Γ *of constraints over the concrete domain* $\mathcal{D}_{\mathbf{L}}$*, a* canonical valuation *of* Γ *w.r.t.* $\mathcal{F}$ *is a valuation* $v_{\Gamma,\mathcal{F}}$ *such that*

1. $v_{\Gamma,\mathcal{F}} \models \mathcal{F}$ *and*
2. $v_{\Gamma,\mathcal{F}} \models \alpha$ *iff.* $\mathcal{D}_{\mathbf{L}}, \mathcal{F} \models \bigsqcap \Gamma \sqsubseteq \alpha$ *for each constraint* α*.*

Moreover, we say that $\mathcal{D}_{\mathbf{L}}$ *has canonical valuations w.r.t.* $\mathcal{F}$ *if such a valuation* $v_{\Gamma,\mathcal{F}}$ *exists for every finite, w.r.t.* $\mathcal{F}$ *satisfiable* Γ*.*

Since for each constraint α in Γ, the inclusion $\bigsqcap \Gamma \sqsubseteq \alpha$ is valid, we infer with the second condition that $v_{\Gamma,\mathcal{F}}$ satisfies Γ.

A semi-lattice $\mathbf{L}$ is *complete* if every subset $P \subseteq L$ has a meet $\bigwedge P \in L$, i.e. such that $\bigwedge P \leq p$ for each $p \in P$ and, if $q \leq p$ for each $p \in P$, then $q \leq \bigwedge P$. Note that these two conditions generalize (SL4) and (SL5).

Theorem 8. *For each complete semi-lattice* $\mathbf{L}$ *and for every FBox* $\mathcal{F}$*, the concrete domain* $\mathcal{D}_{\mathbf{L}}$ *has canonical valuations and so is convex w.r.t.* $\mathcal{F}$*.*

Theorem 9. *Let* $\mathbf{L}$ *be a computable, bounded semi-lattice and* $\mathcal{F}$ *be an FBox. If* $\mathbf{L}$ *is well-founded or* $\mathcal{F}$ *is acyclic, then the concrete domain* $\mathcal{D}_{\mathbf{L}}$ *has computable canonical valuations and is admissible w.r.t.* $\mathcal{F}$*.*

Now, we want to determine the time requirement for computing a canonical valuation $v_{\Gamma,\mathcal{F}}$, which is measured w.r.t. the constraint set Γ and the FBox $\mathcal{F}$.

An operation $H\colon L^n \to L$ is *non-duplicating* if, for all $(p_1, \ldots, p_n) \in L^n$, the size of $H(p_1, \ldots, p_n)$ is no larger than the size of $(p_1, \ldots, p_n)$. An FBox is *non-duplicating* if all operations in it are non-duplicating and each feature occurs at most once in any right-hand side.

Proposition 10. *Consider a polynomial-time computable, bounded semi-lattice* $\mathbf{L}$ *such that its meet operation is non-duplicating. Further consider an acyclic, non-duplicating FBox* $\mathcal{F}$ *in which all occurring operations are polynomial-time computable. W.r.t.* $\mathcal{F}$*, the concrete domain* $\mathcal{D}_{\mathbf{L}}$ *has polynomial-time computable canonical valuations and is* P*-admissible.*

We obtain exponential complexity if $\wedge$ and $\mathcal{F}$ are not non-duplicating.

Proposition 11. *For every polynomial-time computable, bounded semi-lattice* $\mathbf{L}$ *and for every acyclic FBox* $\mathcal{F}$ *in which all occurring operations are polynomial-time computable, the concrete domain* $\mathcal{D}_{\mathbf{L}}$ *has exponential-time computable canonical valuations and is* EXP*-admissible w.r.t.* $\mathcal{F}$*.*

3.1 Intervals

Let N be a non-empty set of real numbers. The semi-lattice $\mathbf{Int}(N)$ consists of all intervals over N, is partially ordered by set inclusion $\subseteq$ and has set intersection $\cap$ as its meet operation. $\mathbf{Int}(N)$ is already bounded since its greatest element is

$N = (-\infty, \infty)$, but we rather identify it with $\square$ and add an artificial greatest element $\top$. It also has a smallest element $\emptyset = (p, p)$ where $p \in N$ is arbitrary, and we identify this smallest element with the contradictory value $\bot$. The inclusion satisfies $[p_1, q_1] \subseteq [p_2, q_2]$ iff. $p_2 \leq p_1$ and $q_1 \leq q_2$, and the intersection satisfies $[p_1, q_1] \cap [p_2, q_2] = [\mathsf{max}(p_1, p_2), \mathsf{min}(q_1, q_2)]$, and similarly for the other interval types. It follows that $\mathbf{Int}(N)$ is polynomial-time computable since $\leq$ is decidable in polynomial time [30], and its meet operation is non-duplicating.

The hierarchical concrete domain $\mathcal{D}_{\mathbf{Int}(N)}$ is called the *interval domain* over N. Since for every number $p \in N$, the singleton $\{p\}$ equals the interval $[p, p]$, we can specify the precise numerical value of a feature with the constraint $f \subseteq \{p\}$, also written $f = p$. Moreover, instead of $f \subseteq [p, q]$ we may also write $p \leq f \leq q$.

Example 12. Through the interval domain over the non-negative 8-bit integers $N := \mathbb{N} \cap [0, 2^8{-}1]$ we could express non-elevated blood pressure by $\mathsf{NonElevatedBP} \equiv (\mathsf{sys} \subseteq [0, 120)) \sqcap (\mathsf{dia} \subseteq [0, 70))$, elevated blood pressure by $\mathsf{ElevatedBP} \equiv (\mathsf{sys} \subseteq [120, 140)) \sqcap (\mathsf{dia} \subseteq [70, 90))$, and hypertension by $(\mathsf{sys} \subseteq [140, \infty)) \sqsubseteq \mathsf{Hypertension}$ and $(\mathsf{dia} \subseteq [90, \infty)) \sqsubseteq \mathsf{Hypertension}$. With the above syntactic sugar, the first statement can also be written as $\mathsf{NonElevatedBP} \equiv (0 \leq \mathsf{sys} < 120) \sqcap (0 \leq \mathsf{dia} < 70)$, and similarly for the other two. The concrete values of patient bob can be represented by the assertions $\mathsf{bob} : (\mathsf{sys} = 114)$ and $\mathsf{bob} : (\mathsf{dia} \subseteq [69, 69])$. The KB consisting of all these aforementioned statements entails $\mathsf{bob} : \mathsf{NonElevatedBP}$.

Example 13. Continuing Example 4, we can additionally consider the two FIs $\mathsf{dia} \subseteq \mathsf{sys} - \mathsf{pp}$ and $\mathsf{sys} \subseteq \mathsf{dia} + \mathsf{pp}$, which allow us to also infer interval values of dia and sys given interval values of the respective other two. Importantly, this does not destroy convexity.

This is in stark contrast to the concrete domain extending $\mathcal{D}_{\mathbb{Q},\mathsf{diff}}$ with constraints $f \geq b$, $f < b$, $f \leq b$, which allows to express interval values as well (in a different way though). There, the constraint inclusion $(\mathsf{sys} - \mathsf{dia} = 40) \sqsubseteq (\mathsf{sys} \leq 120) \sqcup (\mathsf{dia} > 80)$ is valid, violating convexity. Additionally using the expressivity of $\mathcal{D}_{\mathbb{Q},\mathsf{lin}}$, we could express that $\mathsf{pp} = \mathsf{sys} - \mathsf{dia}$ by the CI $\top \sqsubseteq (\mathsf{sys} - \mathsf{dia} - \mathsf{pp} = 0)$ as in Example 3 in [1]. Under this CI, the constraint inclusion $(\mathsf{pp} = 40) \sqsubseteq (\mathsf{sys} \leq 120) \sqcup (\mathsf{dia} > 80)$ would be valid, also violating convexity.

In our interval domain over the non-negative integers and with the cyclic FBox $\{\mathsf{pp} \subseteq \mathsf{sys} - \mathsf{dia},\ \mathsf{dia} \subseteq \mathsf{sys} - \mathsf{pp},\ \mathsf{sys} \subseteq \mathsf{dia} + \mathsf{pp}\}$, the similar constraint inclusion $(\mathsf{pp} \subseteq [40, 40]) \sqsubseteq (\mathsf{sys} \subseteq [0, 120]) \sqcup (\mathsf{dia} \subseteq (80, \infty))$ is not valid. A countervaluation is v where $v(\mathsf{sys}) = [40, \infty)$, $v(\mathsf{dia}) = [0, \infty)$, $v(\mathsf{pp}) = [40, 40]$. It satisfies the first FI since $[40, \infty) - [0, \infty) = [0, \infty) \supseteq [40, 40]$, the second FI since $[40, \infty) - [40, 40] = [0, \infty) \supseteq [0, \infty)$, and the third FI since $[0, \infty) + [40, 40] = [40, \infty) \supseteq [40, \infty)$.

Recall that the interval semi-lattice $\mathbf{Int}(N)$ is defined for every non-empty set N of real numbers. The set N is partially ordered by the usual ordering $\leq$ and has the meet operation min, i.e. $(N, \leq, \mathsf{min})$ is itself a semi-lattice. It thus makes sense to say that N is complete. The real numbers $\mathbb{R}$, the non-negative real numbers $\mathbb{R}_+$, the integers $\mathbb{Z}$, the natural numbers $\mathbb{N}$, the n-bit integers, the n-bit floating-point numbers, the n-bit fixed-point numbers, and all finite subsets of $\mathbb{R}$ are complete, but the rational numbers $\mathbb{Q}$ is not — for instance, the infimum of $\{ (1 + 1/n)^{n+1} \mid n \geq 0 \}$ is Euler's number e, an irrational number. It is easy to see that the semi-lattice $\mathbf{Int}(N)$ is complete if the number set N is complete, and so we obtain the below corollary to Theorem 8.

Corollary 14. *If the semi-lattice* $(N, \leq, \mathsf{min})$ *is complete, then the interval domain* $\mathcal{D}_{\mathbf{Int}(N)}$ *has canonical valuations and is convex w.r.t. every FBox* $\mathcal{F}$*.*

An immediate consequence of Theorem 9 is that the interval domain $\mathcal{D}_{\mathbf{Int}(\mathbb{R})}$ over all real numbers is admissible w.r.t. every acyclic FBox. Moreover, an obvious corollary to Proposition 10 is as follows.

Corollary 15. *W.r.t. each acyclic, non-duplicating FBox* $\mathcal{F}$ *in which all operations are polynomial-time computable, the interval domain* $\mathcal{D}_{\mathbf{Int}(\mathbb{R})}$ *has polynomial-time-computable canonical valuations and is* P*-admissible.*

Next, we employ linear programming to handle affine FBoxes, which might be cyclic. We call an FBox $\mathcal{F}$ *affine* if all operations in FIs in $\mathcal{F}$ are affine, i.e. all FIs are of the form $f \subseteq \sum_{i=1}^{n} P_i \cdot g_i + Q_i$ where the P_i and Q_i are intervals. For instance, the FI $\mathsf{pp} \subseteq \mathsf{sys} - \mathsf{dia}$ is affine, but $\mathsf{bmi} \subseteq \mathsf{bodyMass}/\mathsf{bodyHeight}^2$ is not. Since each affine FI represents two linear inequalities (one for the lower bound of the interval value of f, and another one for the upper bound), we can transform affine FBoxes into linear programs, which can be solved in polynomial time [33]. We thus obtain the following result.

Proposition 16. *Let* $\underline{c}, \overline{c} \in \mathbb{R}_{+}$ *be non-negative real numbers such that* $\underline{c} \leq \overline{c}$*. Restricted to closed intervals only, the interval domain* $\mathcal{D}_{\mathbf{Int}([\underline{c},\overline{c}])}$ *over the non-negative real numbers between* $\underline{c}$ *and* $\overline{c}$ *is* P*-admissible w.r.t. each affine FBox* $\mathcal{F}$*, i.e. all FIs are of the form* $f \subseteq \sum_{i=1}^{n} [\underline{a}_i, \overline{a}_i] \cdot g_i + [\underline{b}, \overline{b}]$*.*

It remains an open problem, whether the interval domains $\mathcal{D}_{\mathbf{Int}([\underline{c},\overline{c}])}$ remain P-admissible w.r.t. affine FBoxes when all interval types would be considered. We conjecture that the interval bounds can be computed using the same linear program, but determining the correct interval types (closed or open at the lower bound, closed or open at the upper bound) could possibly lead to a combinatorial explosion. It is further unclear whether, without the bounding interval $[\underline{c}, \overline{c}]$, the interval domain $\mathcal{D}_{\mathbf{Int}(\mathbb{R}_{+})}$ would still be P-admissible w.r.t. affine FBoxes. The canonical valuation could then send features to intervals with upper bound $+\infty$, in which case the polytope described by the inequations would be unbounded. This requires an LP-solver with support for unbounded solution polytopes.

We can also handle affine FBoxes together with negative numbers, but then need to restrict the coefficient intervals $[\underline{a}_i, \overline{a}_i]$ to singletons — as otherwise the non-linear functions min and max would be required to compute a product $[\underline{a}_i, \overline{a}_i] \cdot g_i$, i.e. the system of inequalities would not be linear anymore and could therefore not be solved by linear-programming methods.

Proposition 17. *Let* $\underline{c}, \overline{c} \in \mathbb{R}$ *be real numbers such that* $\underline{c} \leq \overline{c}$*. Restricted to closed intervals, the interval domain* $\mathcal{D}_{\mathbf{Int}([\underline{c},\overline{c}])}$ *over the real numbers in* $[\underline{c}, \overline{c}]$ *is*

P*-admissible w.r.t. each affine FBox $\mathcal{F}$ involving only singleton coefficients, i.e. all FIs are of the form $f \subseteq \sum_{i=1}^{n} \{a_i\} \cdot g_i + [\underline{b}, \overline{b}]$.*

Linear programming becomes NP-hard when restricted to integers only [38]. Unless $\mathsf{P} = \mathsf{NP}$, the integer interval domains $\mathcal{D}_{\mathbf{Int}(\mathbb{Z})}$, $\mathcal{D}_{\mathbf{Int}(\mathbb{N})}$, and $\mathcal{D}_{\mathbf{Int}(\{0,1\})}$ are thus not P-admissible w.r.t. affine FBoxes. These domains are rather suitable for integration into Horn logics [55] that do not allow for polynomial-time reasoning, such as $\mathcal{ELI}$ [4], Horn-$\mathcal{ALC}$ [47], Horn-$\mathcal{SROIQ}$ [57], and existential rules [15].

Example 18. Example 3 in [1] shows that the combination of the concrete domains $\mathcal{D}_{\mathbb{Q},\mathsf{diff}}$ and $\mathcal{D}_{\mathbb{Q},\mathsf{lin}}$ is not enough to express that intensive-care patients need attention if their pulse pressure is larger than 50 or their current heart rate exceeds their maximal heart rate. Moreover, this combination is not even convex.

With our interval domain these statements can be expressed through the affine FIs $\mathsf{pp} \subseteq \mathsf{sys} - \mathsf{dia}$, and $\mathsf{maxHR} \subseteq 220 - \mathsf{age}$, and $\mathsf{exceedHR} \subseteq \mathsf{hr} - \mathsf{maxHR}$, as well as the CIs $\mathsf{ICUPatient} \sqsubseteq (\mathsf{hr} \subseteq \square) \sqcap (\mathsf{sys} \subseteq \square) \sqcap (\mathsf{dia} \subseteq \square)$, and $\mathsf{ICUPatient} \sqcap (\mathsf{pp} \subseteq (50, \infty)) \sqsubseteq \mathsf{NeedsAttention}$, and $\mathsf{ICUPatient} \sqcap (\mathsf{exceedHR} \subseteq (0, \infty)) \sqsubseteq \mathsf{NeedsAttention}$.

3.2 2D-Polygons

A *2D-polygon* is a finite sequence of successively connected finite line segments in the real plane $\mathbb{R}^2$ such that the end vertex of the last segment equals the start vertex of the first. These line segments form a simple closed curve in $\mathbb{R}^2$, and by the Jordan Curve Theorem [28, 36] each 2D-polygon has an *interior region* (bounded by the curve) and an *exterior region*. In the following we identify each 2D-polygon with the subset of $\mathbb{R}^2$ consisting of its boundary and the interior region. 2D-polygons are thoroughly studied in Computational Geometry and frequently used in geographic information systems (GIS).

Every 2D-polygon can be represented as a finite sequence of vertex coordinates in $\mathbb{R}^2$ — its line segments then connect each two subsequent coordinates and, respectively, the first and last coordinate — and thus deciding the set of all 2D-polygons is trivial. Clipping algorithms allow for deciding in polynomial time if a polygon is a subset of another (i.e. polygon containment without moving or scaling operations) as well as for computing any Boolean operation involving two polygons (union, intersection, difference, xor) in polynomial time [24, 54, 63]. However, intersections can be of quadratic size and might consist of unions of disjoint 2D-polygons. In order to obtain a semi-lattice, which must be closed under its meet operation, it would therefore be necessary to take the set of all finite unions of separated 2D-polygons: we denote it by $\mathbf{UGon}(\mathbb{R}^2)$, its partial order is containment $\subseteq$, and its meet is intersection $\cap$. According to the above references, $\mathbf{UGon}(\mathbb{R}^2)$ is polynomial-time computable (w.r.t. arithmetic complexity). The hierarchical concrete domain $\mathcal{D}_{\mathbf{UGon}(\mathbb{R}^2)}$ is called *polygon domain* over $\mathbb{R}^2$. A corollary to Proposition 11 is as follows.

Corollary 19. *W.r.t. arithmetic complexity, the polygon domain $\mathcal{D}_{\mathbf{UGon}(\mathbb{R}^2)}$ has exponential-time computable canonical valuations and is* EXP*-admissible w.r.t. each acyclic FBox $\mathcal{F}$ in which all operations are polynomial-time computable.*

To the best of the author's knowledge, it is unclear whether the intersection of n polygons might reach an exponential size. If this worst case would not be possible and, moreover, all operations in $\mathcal{F}$ are non-duplicating, then $\mathcal{D}_{\mathbf{UGon}(\mathbb{R}^2)}$ would even be P-admissible w.r.t. $\mathcal{F}$ (w.r.t. arithmetic complexity).

Example 20. Locations can be represented as polygons in the real plane $\mathbb{R}^2$. For instance, we have "Nöthnitzer Straße 46, 01187 Dresden" $\subseteq$ "01187 Dresden" $\subseteq$ "Dresden" $\subseteq$ "Saxony" $\subseteq$ "Germany" $\subseteq$ "Europe" $\subseteq$ "Earth".

The situation is computationally easier with *convex* 2D-polygons, which contain all line segments between each two of their points. One can think of convex 2D-polygons as two-dimensional generalizations of closed intervals. Both in linear time, we can decide the subset relation $\subseteq$ and compute the intersection operation $\cap$ for convex 2D-polygons [56, 60, 62]. Intersection is non-duplicating [60]. However, deciding the set of all convex 2D-polygons is not trivial anymore but needs linear time [60]. We denote the semi-lattice of all convex 2D-polygons by $\mathbf{CGon}(\mathbb{R}^2)$, and it is linear-time computable (w.r.t. arithmetic complexity). The hierarchical concrete domain $\mathcal{D}_{\mathbf{CGon}(\mathbb{R}^2)}$ is called *convex-polygon domain* over $\mathbb{R}^2$.

Obviously, convex polygons are closed under intersection but not under union, difference, and xor. Since union is monotonic, it can be used in FBoxes when followed by the convex-hull operation (which computes the smallest enclosing polygon that is convex). This is, however, not possible for difference and xor since they are not monotonic. Suitable monotonic operations besides intersection and convex union are translation, rotation, and scaling, and these can be computed in linear time as well. Below is a corollary to Proposition 10.

Corollary 21. *W.r.t. each acyclic, non-duplicating FBox $\mathcal{F}$ in which all occurring operations are polynomial-time computable, the convex-polygon domain $\mathcal{D}_{\mathbf{CGon}(\mathbb{R}^2)}$ has polynomial-time computable canonical valuations and is* P-*admissible (w.r.t. arithmetic complexity).*

Contrary to $\mathbf{Int}(\mathbb{R})$, neither $\mathbf{UGon}(\mathbb{R}^2)$ nor $\mathbf{CGon}(\mathbb{R}^2)$ are complete. One reason is that the unit circle can be obtained as the intersection of regular polygons (for each $n \in \mathbb{N}$ with $n \geq 3$, take a smallest regular n-sided polygon that encloses the unit circle). The polygon semi-lattices are also not well-founded, and thus we cannot obtain corollaries to Theorems 8 and 9 w.r.t. cyclic FBoxes.

3.3 Regular Languages

Given a finite alphabet Σ, the semi-lattice $\mathbf{Reg}(\Sigma)$ consists of all regular languages over Σ, is partially ordered by set inclusion $\subseteq$, and its meet operation is set intersection $\cap$. It is not complete since regular languages are not closed under arbitrary intersections (only under finite ones). More specifically, $L = \bigcap\{\, \Sigma^* \setminus \{w\} \mid w \notin L \,\}$ for each language L, and thus for two symbols $a, b \in \Sigma$ the non-regular language $\{\, a^n b^n \mid n \in \mathbb{N} \,\}$ is an intersection of regular languages. Thus, convexity does not follow from Theorem 8.

In order to obtain a computable semi-lattice, we need to work with finite representations of regular languages. With regular expressions, binary intersections of regular languages can have exponential size even over a binary alphabet [25], i.e. the meet would not be computable in polynomial time. It is no alternative to instead use one-unambiguous/deterministic regular expressions since they cannot describe all regular languages and are not even closed under intersection, even though their inclusion problem is in polynomial time [19, 32, 48].

Using finite automata as representations is preferred, on the one hand since to obtain the meet/intersection of two regular languages we can compute the product of the respective finite automata in polynomial time [37]. On the other hand, a language inclusion $L_1 \subseteq L_2$ holds iff. the language equivalence $L_1 \cap L_2 = L_2$ holds, and thus it suffices to check if the product of both finite automata is equivalent to the second automaton. For deterministic automata this is possible in polynomial time [16, 31], but otherwise needs polynomial space [61].

The semi-lattice $\mathbf{DFA}(\Sigma)$ consists of all deterministic finite automata over Σ, is partially ordered by automata inclusion $\preceq$ where $\mathfrak{A} \preceq \mathfrak{B}$ if $L(\mathfrak{A}) \subseteq L(\mathfrak{B})$, and its meet operation is the product $\times$, which satisfies $L(\mathfrak{A} \times \mathfrak{B}) = L(\mathfrak{A}) \cap L(\mathfrak{B})$. It is thus polynomial-time computable. Furthermore, $\mathbf{FA}(\Sigma)$ comprises all finite automata and is polynomial-space computable. Since finite automata and deterministic ones have equal power in the sense that they both describe all regular languages, both semi-lattices can serve as representations of $\mathbf{Reg}(\Sigma)$.

The hierarchical concrete domains $\mathcal{D}_{\mathbf{DFA}(\Sigma)}$ and $\mathcal{D}_{\mathbf{FA}(\Sigma)}$ are called the *regular-language domains* over Σ. Since single words are regular languages, precise string values are supported: we may write $(f = w)$ instead of $(f \preceq \mathfrak{A})$ when $L(\mathfrak{A}) = \{w\}$. Further note that $\Box$ is the automaton that accepts every string, $\bot$ accepts no string at all, and $\top$ is an artificial greatest element.

Example 22. Let Σ be an alphabet containing all Latin letters, e.g. The Unicode Standard. We use a feature $\mathsf{hasTitle}$ to represent the title string of a research paper. Further take a DFA $\mathfrak{A}$ such that $L(\mathfrak{A}) = \Sigma^* \circ \{\text{description logic}\} \circ \Sigma^*$. With that, the CI $\mathsf{ScientificArticle} \sqcap (\mathsf{hasTitle} \preceq \mathfrak{A}) \sqsubseteq \mathsf{DLPaper}$ expresses that the concept of all DL papers subsumes the concept of all scientific articles with a title containing "description logic" as substring.

Even without an FBox, the regular-language domains $\mathcal{D}_{\mathbf{DFA}(\Sigma)}$ and $\mathcal{D}_{\mathbf{FA}(\Sigma)}$ are in general not P-admissible. In a nutshell, meets need not be non-duplicating, and thus accumulating all upper bounds of the same feature could yield an exponentially large automaton. More specifically, if a constraint set Γ contains several constraints $f \leq \mathfrak{A}$ for the same feature f, then computing the value $v_{\Gamma,\mathcal{F}}(f)$ boils down to computing the intersection of all these automata $\mathfrak{A}$. Since emptiness of intersections of finite automata is PSpace-hard [43] and graph reachability is NL-complete [35], $v_{\Gamma,\mathcal{F}}(f)$ cannot be computed in polynomial time, unless $\mathsf{P} = \mathsf{PSpace}$. We obtain, however, the following corollary to Proposition 11.

Corollary 23. *W.r.t. each acyclic FBox $\mathcal{F}$ in which all occurring operations are polynomial-time computable, the regular-language domain $\mathcal{D}_{\mathbf{DFA}(\Sigma)}$ has exponential-time computable canonical valuations and is EXP-admissible.*

The DFA operations corresponding to the language operations union $\cup$, intersection $\cap$, and complement $^-$ are polynomial-time computable. $\mathcal{D}_{\mathbf{DFA}(\Sigma)}$ is thus EXP-admissible w.r.t. each acyclic FBox involving these operations only. In contrast, concatenation $\circ$, Kleene-star *, mirror/reversal $^{\leftarrow}$, left-quotients $\backslash$, and right-quotients $/$ on DFAs are exponential-time computable but not polynomial-time computable [65]. However on FAs, all operations but complement are polynomial-time computable, and mirror/reversal is even non-duplicating. $\mathcal{D}_{\mathbf{FA}(\Sigma)}$ is $\mathsf{EXPSpace}$-admissible w.r.t. acyclic FBoxes using these polynomial-time operations.

It is worth mentioning that, if we have at most one inclusion (i.e. constraint or FI) per feature, then in the procedure in the proof of Theorem 9 neither the automata product operation nor the automata inclusion relation needs to be used, and so we have the following corollary.

Corollary 24. *Let $\mathcal{F}$ be an acyclic, non-duplicating FBox in which all occurring operations are polynomial-time computable. Further let Γ be a constraint set. If $\mathcal{F} \cup \Gamma$ contains, for each feature f, at most one inclusion with f on the left, then the canonical valuation of Γ w.r.t. $\mathcal{F}$ can be computed in polynomial time.*

Example 25. Assume the features $\mathsf{givenName}$, $\mathsf{familyName}$, and name are used to represent persons' names. Then for instance, the concept $\mathsf{Male} \sqcap (\mathsf{givenName} \preceq \mathfrak{A})$ where $L(\mathfrak{A}) = \{\mathrm{F}\} \circ \Sigma^*$ describes all males whose given name starts with 'F'.

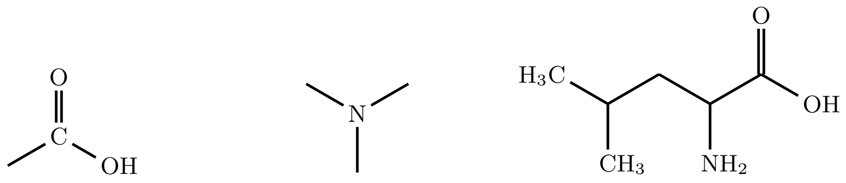
Moreover, the FI $\mathsf{name} \preceq \mathsf{givenName} \circ \{\_\} \circ \mathsf{familyName}$ allows to infer a regular language value of name when values of $\mathsf{givenName}$ and $\mathsf{familyName}$ are available (i.e. both are not $\top$). If the latter two are precise values (languages consisting of a single word), then also name gets a precise value through the FI. Note that '_' stands for a white space. The FI $\mathsf{shortName} \preceq \mathsf{initial}(\mathsf{givenName}) \circ \{.\_\} \circ \mathsf{familyName}$ generates a shortened form of a name that only contains the initial of the given name followed by a dot, where the function $\mathsf{initial}$ is defined by $L(\mathsf{initial}(\mathfrak{A})) := \{\, s \mid s \in \Sigma \text{ and there is } w \in \Sigma^* \text{ such that } s \circ w \in L(\mathfrak{A}) \,\}$.

The semi-lattices $\mathbf{Reg}(\Sigma)$, $\mathbf{DFA}(\Sigma)$, and $\mathbf{FA}(\Sigma)$ are not well-founded since, already over the unary alphabet $\{a\}$, the regular languages $L_i := \{\, a^j \mid i \leq j \,\}$ where $i \in \mathbb{N}$ form an infinite descending chain $L_0 \supset L_1 \supset L_2 \supset \cdots$. These semi-lattices are also not complete (see above). W.r.t. cyclic FBoxes, we can thus not conclude convexity by Theorems 8 and 9.

For a restricted class of FBoxes, however, we obtain systems of language inclusions known to be solvable in exponential time [10]. An n-ary operation H on $\mathbf{DFA}(\Sigma)$ is *left-linear* if $H(\mathfrak{X}_1, \ldots, \mathfrak{X}_n) = \mathfrak{X}_1 \circ \mathfrak{A}_1 \cup \cdots \cup \mathfrak{X}_n \circ \mathfrak{A}_n \cup \mathfrak{B}$ and *right-linear* if $H(\mathfrak{X}_1, \ldots, \mathfrak{X}_n) = \mathfrak{A}_1 \circ \mathfrak{X}_1 \cup \cdots \cup \mathfrak{A}_n \circ \mathfrak{X}_n \cup \mathfrak{B}$, where $\mathfrak{A}_1, \ldots, \mathfrak{A}_n, \mathfrak{B}$ are DFAs. An FBox $\mathcal{F}$ is *linear* if the operations in its FIs are either all left-linear or all right-linear.

Proposition 26. *The regular-language domain $\mathcal{D}_{\mathbf{DFA}(\Sigma)}$ has exponential-time computable canonical valuations and is EXP-admissible w.r.t. each linear FBox.*

If precise values (single words) are sufficient for the application, we could also use the semi-lattice $(\Sigma^* \cup \{\bot, \top\}, \leq, \wedge)$ where $\leq$ is the smallest partial order such



(a) $\mathcal{G}_{\text{carboxylic acid group}}$ (b) $\mathcal{G}_{\text{amino group}}$ (c) $\mathcal{G}_{\text{L-leucine}}$

Fig. 1: Three graphs representing chemical compounds

that $\bot < w < \top$ for each $w \in \Sigma^*$. The meet operation $\wedge$ thus satisfies $\top \wedge w = w$, $w \wedge w = w$, and $w \wedge \bot = \bot$ for each $w \in \Sigma^* \cup \{\bot, \top\}$, and $w_1 \wedge w_2 = \bot$ whenever $w_1, w_2 \in \Sigma^*$ with $w_1 \neq w_2$. This semi-lattice is complete and, by Theorem 8, its hierarchical concrete domain is convex w.r.t. every FBox. Since during the computation of a canonical valuation each feature value can be refined at most two times (from $\top$ to some w, and then possibly to $\bot$), this concrete domain is P-admissible w.r.t. each FBox in which all operations are polynomial-time computable. The disadvantage is, however, that string search like in Example 22 is not possible anymore. On the other hand, this suggests that in $\mathcal{D}_{\mathbf{DFA}(\Sigma)}$ and $\mathcal{D}_{\mathbf{FA}(\Sigma)}$ everything involving only precise values is possible in polynomial time.

3.4 Graphs

All finite, labeled graphs constitute a semi-lattice **Graph**, where the partial order $\leq$ is defined by $\mathcal{G} \leq \mathcal{H}$ if there is a homomorphism from $\mathcal{H}$ to $\mathcal{G}$. It is well-known that $\leq$ is NP-complete [21], but in P for acyclic graphs [22]. The meet of two graphs is their disjoint union, thus a non-duplicating operation, and the greatest element in this semi-lattice is the empty graph. Obviously, **Graph** is neither complete nor well-founded, and so we cannot apply Theorems 8 and 9. It thus remains unclear whether the *graph domain* $\mathcal{D}_{\mathbf{Graph}}$ is convex w.r.t. cyclic FBoxes.

Corollary 27. *The graph domain* $\mathcal{D}_{\mathbf{Graph}}$ *has computable canonical valuations w.r.t. acylic FBoxes. Moreover, it is* NP*-admissible w.r.t. every acyclic, non-duplicating FBox in which all operations are polynomial-time computable, and it is* EXP*-admissible w.r.t. every acyclic FBox in which all operations are polynomial-time computable.*

Example 28. Structural formulas of molecules can be represented as labeled graphs. Each node is labeled with the atom it represents, and the edges are labeled with the binding type (e.g. single bond, double bond, etc.). Figure 1 shows three exemplary graphs. Graph (c) represents L-leucine, and we can integrate it into a KB with the statement $\mathsf{L\text{-}Leucine} \equiv (\mathsf{hasMolecularStructure} \leq \mathcal{G}_{\text{L-leucine}})$. Moreover, the statement $\mathsf{AminoAcid} \equiv (\mathsf{hasMolecularStructure} \leq \mathcal{G}_{\text{carboxylic acid group}}) \sqcap (\mathsf{hasMolecularStructure} \leq \mathcal{G}_{\text{amino group}})$ expresses that amino acids are organic compounds that contain both amino and carboxylic acid functional groups. If $\mathcal{K}$ is the KB consisting of the aforementioned statements, then $\mathcal{K} \models \mathsf{L\text{-}Leucine} \sqsubseteq \mathsf{AminoAcid}$ since $\mathcal{G}_{\text{L-leucine}} \leq \mathcal{G}_{\text{carboxylic acid group}} \wedge \mathcal{G}_{\text{amino group}}$.

4 Reasoning in $\mathcal{EL}^{++}$ with Hierarchical Concrete Domains

Like other convex concrete domains, a hierarchical concrete domain $\mathcal{D}_{\mathbf{L}}$ can be integrated into $\mathcal{EL}^{++}$ but, in addition to Section 2, every $\mathcal{EL}^{++}[\mathcal{D}_{\mathbf{L}}]$ KB may contain finitely many FIs. Of course, a model of such a KB must also satisfy all FIs in it. In order to guarantee that reasoning is decidable, a restriction on the interplay of RIs and range inclusions must be fulfilled by every $\mathcal{EL}^{++}[\mathcal{D}]$ KB [5], see Condition 1 below. To this end, we define the *range set* of a role r in $\mathcal{K}$ by $\mathsf{Range}(r,\mathcal{K}) := \{\, C \mid \text{there is a role } s \text{ s.t. } \mathcal{R} \models r \sqsubseteq s \text{ and } \mathsf{Ran}(s) \sqsubseteq C \in \mathcal{K} \,\}$, where $\mathcal{R}$ is the subset of all RIs in $\mathcal{K}$. All such range sets can be computed in polynomial time by first transforming each RI $r_1 \circ \cdots \circ r_n \sqsubseteq s$ into a context-free grammar rule $s \to r_1 \ldots r_n$ (see Lemma IV in [9] for details) and then deciding the word problem for this grammar (e.g. with the CYK algorithm [23, 39, 64]).

Definition 29. *Consider a bounded semi-lattice* $\mathbf{L}$. *An* $\mathcal{EL}^{++}[\mathcal{D}_{\mathbf{L}}]$ knowledge base (KB) $\mathcal{K}$ *is a finite set of CIs, RIs, range inclusions, and FIs such that*

1. $\mathsf{Range}(s,\mathcal{K}) \subseteq \mathsf{Range}(r_n,\mathcal{K})$ *for every RI* $r_1 \circ \cdots \circ r_n \sqsubseteq s$ *in* $\mathcal{K}$ *with* $n \geq 2$,
2. *and the hierarchical concrete domain* $\mathcal{D}_{\mathbf{L}}$ *is convex w.r.t. all FIs in* $\mathcal{K}$.

For a complexity class C *we say that* $\mathcal{D}_{\mathbf{L}}$ *is* C-admissible *w.r.t.* $\mathcal{K}$ *if* $\mathcal{D}_{\mathbf{L}}$ *is* C-*admissible w.r.t. the FBox consisting of all FIs in* $\mathcal{K}$.

For Condition 1 range inclusions on s must not imply further concept memberships than already implied by the range inclusions on r_n; otherwise emptiness of intersections of two context-free grammars could be reduced to subsumption [5].

Reasoning in $\mathcal{EL}^{++}[\mathcal{D}]$ can be done by means of a rule-based calculus [4, 5, 40, 42], and a hierarchical concrete domain $\mathcal{D}_{\mathbf{L}}$ can be seamlessly integrated into this calculus. Compared to the primal calculus [4, 5], it is only necessary to take the FIs into account. For integration into the improved calculus [40, 42] we only need to add the following two rules responsible for interaction between concrete and logical reasoning (where $\mathcal{F}$ consists of all FIs in the KB), see [46] for details.

$$\mathsf{R}_{\mathcal{D}}\colon \quad \frac{C \sqsubseteq (f_1 \leq p_1) \ \cdots \ C \sqsubseteq (f_m \leq p_m)}{C \sqsubseteq (g \leq q)} : \mathcal{D}_{\mathbf{L}}, \mathcal{F} \models \prod_{i=1}^{m} (f_i \leq p_i) \sqsubseteq (g \leq q)$$

$$\mathsf{R}_{\mathcal{D},\bot}\colon \quad \frac{C \sqsubseteq (f_1 \leq p_1) \ \cdots \ C \sqsubseteq (f_m \leq p_m)}{C \sqsubseteq \bot} : \prod_{i=1}^{m} (f_i \leq p_i) \text{ unsatisfiable in } \mathcal{D}_{\mathbf{L}}, \mathcal{F}$$

However, we restrict attention to nominal-safe KBs, i.e. nominals $\{i\}$ must not occur in conjunctions and each right-hand side of a concept or range inclusion must not be a single nominal $\{i\}$. Full support for nominals in $\mathcal{EL}^{++}[\mathcal{D}]$ is technically quite involved and makes reasoning more expensive: the degree of the polynomial describing the worst-case reasoning time would then be larger by 1 [41]. We conjecture the same for $\mathcal{EL}^{++}[\mathcal{D}_{\mathbf{L}}]$ KBs that are not nominal-safe.

Range inclusions are not natively supported by the rule-based calculus, but they must rather be eliminated [5]. This transformation was originally described for KBs in normal form only, but can now be done without prior transformation to normal form, see [46] for details.

Theorem 30. *Let $\mathbf{L}$ be a bounded semi-lattice. For all nominal-safe $\mathcal{EL}^{++}[\mathcal{D}_\mathbf{L}]$ KBs w.r.t. which the hierarchical concrete domain $\mathcal{D}_\mathbf{L}$ is P-admissible, the following reasoning tasks can be done in polynomial time: consistency, classification, subsumption checking, instance checking, and concept satisfiability.*

In the proof of the above result, we build a canonical model of the KB iff. it is consistent. Now with the hierarchical concrete domains we can use the canonical valuations for this. The benefit is that the canonical model is universal w.r.t. all nominal-safe assertions $\{i\} \sqsubseteq C$, before it was only universal w.r.t. such assertions without concrete constraints. Our canonical models are thus appropriate for computing optimal repairs [8, 9, 44, 45] of KBs involving concrete domains.

We can also use NP- or EXP-admissible concrete domains in $\mathcal{EL}^{++}$. Reasoning works in the very same way, i.e. the logical reasoning can still be done in polynomial time, but the concrete reasoning is more expensive.

Theorem 31. *Fix a bounded semi-lattice $\mathbf{L}$. For all nominal-safe $\mathcal{EL}^{++}[\mathcal{D}_\mathbf{L}]$ KBs w.r.t. which the hierarchical concrete domain $\mathcal{D}_\mathbf{L}$ is NP-admissible, the following reasoning problems are in NP: consistency, concept satisfiability, subsumption checking, and instance checking. They are in EXP if $\mathcal{D}_\mathbf{L}$ is EXP-admissible. In both cases, the classification can be computed in exponential time.*

5 Future Prospects

An interesting question for future research is whether non-local feature inclusions $f \leq H(R_1 \circ g_1, \ldots, R_n \circ g_n)$ would lead to undecidability or could be reasoned with, where the R_i are role chains. The operator must then be defined for lists of values, like in the non-local feature inclusion $\mathsf{combinedWealth} \subseteq \sum(\mathsf{hasAccount} \circ \mathsf{balance}) + \sum(\mathsf{holdsAsset} \circ \mathsf{value})$ over the interval domain, which computes the aggregated wealth of a person or company. At first sight, it seems that the undecidability proof for $\mathcal{EL}(\mathcal{D}_{\mathbb{Q}^2,\mathsf{aff}})$ [13] cannot be adapted to this setting. (Mind the braces: $(\mathcal{D})$ instead of $[\mathcal{D}]$ allows for role chains in front of features.) The computation of canonical valuations must then take into account the graph structure induced by the role assertions entailed by the knowledge base.

In general, it is unclear whether a hierarchical concrete domain is admissible w.r.t. cyclic FBoxes. According to our results for interval domains and regular-language domains, admissibility can be ensured by approaches to solving systems of equations or inequations involving elements of the underlying semi-lattice. This is still open for the polygon domains and the graph domains.

Since hierarchical concrete domains are convex by design, they are also appropriate for other Horn logics and existential rules — extending the chase procedure with support for them would be practically relevant.

Acknowledgments This work has been supported by Deutsche Forschungsgemeinschaft (DFG) in Projects 389792660 and 558917076 as well as by the Saxon State Ministry for Science, Culture, and Tourism (SMWK) by funding the Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI).

References

1. Alrabbaa, C., Baader, F., Borgwardt, S., Koopmann, P., Kovtunova, A.: Combining Proofs for Description Logic and Concrete Domain Reasoning. In: Proceedings of the 7th International Joint Conference on Rules and Reasoning (RuleML+RR). LNCS, vol. 14244, pp. 54–69. Springer, Heidelberg (2023). `https://doi.org/10.1007/978-3-031-45072-3_4`
2. Baader, F., Borgwardt, S., Lippmann, M.: Query Rewriting for DL-Lite with n-ary Concrete Domains. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI), pp. 786–792 (2017). `https://doi.org/10.24963/IJCAI.2017/109`
3. Baader, F., Bortoli, F.D.: The Abstract Expressive Power of First-Order and Description Logics with Concrete Domains. In: Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing (SAC), pp. 754–761 (2024). `https://doi.org/10.1145/3605098.3635984`
4. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ Envelope. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI), pp. 364–369 (2005). `http://ijcai.org/Proceedings/05/Papers/0372.pdf`
5. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ Envelope Further. In: Proceedings of the 4th OWLED Workshop on OWL: Experiences and Directions. CEUR Workshop Proceedings (2008). `https://ceur-ws.org/Vol-496/owled2008dc%5C_paper%5C_3.pdf`
6. Baader, F., De Bortoli, F.: Logics with Concrete Domains: First-Order Properties, Abstract Expressive Power, and (Un)Decidability. SIGAPP Applied Computing Review **24**(3), 5–17 (2024). `https://doi.org/10.1145/3699839.3699840`
7. Baader, F., Hanschke, P.: A Scheme for Integrating Concrete Domains into Concept Languages. In: Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI), pp. 452–457 (1991). `http://ijcai.org/Proceedings/91-1/Papers/070.pdf`
8. Baader, F., Koopmann, P., Kriegel, F., Nuradiansyah, A.: Computing Optimal Repairs of Quantified ABoxes w.r.t. Static $\mathcal{EL}$ TBoxes. In: Proceedings of the 28th International Conference on Automated Deduction (CADE). LNCS, vol. 12699, pp. 309–326. Springer, Heidelberg (2021). `https://doi.org/10.1007/978-3-030-79876-5_18`
9. Baader, F., Kriegel, F.: Pushing Optimal ABox Repair from $\mathcal{EL}$ Towards More Expressive Horn-DLs. In: Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning (KR), pp. 22–32 (2022). `https://doi.org/10.24963/kr.2022/3`
10. Baader, F., Küsters, R.: Unification in a Description Logic with Transitive Closure of Roles. In: Proceedings of the 8th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR). LNCS, vol. 2250, pp. 217–232. Springer, Heidelberg (2001). `https://doi.org/10.1007/3-540-45653-8_15`

11. Baader, F., Rydval, J.: An Algebraic View on p-Admissible Concrete Domains for Lightweight Description Logics. In: Proceedings of the 17th European Conference on Logics in Artificial Intelligence (JELIA). LNCS, vol. 12678, pp. 194–209. Springer, Heidelberg (2021). `https://doi.org/10.1007/978-3-030-75775-5_14`
12. Baader, F., Rydval, J.: Description Logics with Concrete Domains and General Concept Inclusions Revisited. In: Proceedings of the 10th International Joint Conference on Automated Reasoning (IJCAR). LNCS, vol. 12166, pp. 413–431. Springer, Heidelberg (2020). `https://doi.org/10.1007/978-3-030-51074-9_24`
13. Baader, F., Rydval, J.: Using Model Theory to Find Decidable and Tractable Description Logics with Concrete Domains. Journal of Automated Reasoning **66**(3), 357–407 (2022). `https://doi.org/10.1007/S10817-022-09626-2`
14. Baader, F., Sattler, U.: Description logics with aggregates and concrete domains. Inf. Syst. **28**(8), 979–1004 (2003). `https://doi.org/10.1016/S0306-4379(03)00003-6`
15. Baget, J., Leclère, M., Mugnier, M., Salvat, E.: Extending Decidable Cases for Rules with Existential Variables. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI), pp. 677–682 (2009). `http://ijcai.org/Proceedings/09/Papers/118.pdf`
16. Bonchi, F., Pous, D.: Checking NFA equivalence with bisimulations up to congruence. In: Proceedings of the 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), pp. 457–468 (2013). See also [29] and [17]. `https://doi.org/10.1145/2429069.2429124`
17. Bonchi, F., Pous, D.: Hacking nondeterminism with induction and coinduction. Communications of the ACM **58**(2), 87–95 (2015). `https://doi.org/10.1145/2713167`
18. Borgwardt, S., Bortoli, F.D., Koopmann, P.: The Precise Complexity of Reasoning in $\mathcal{ALC}$ with ω-Admissible Concrete Domains. In: Proceedings of the 37th International Workshop on Description Logics (DL). CEUR Workshop Proceedings (2024). `https://ceur-ws.org/Vol-3739/paper-1.pdf`
19. Brüggemann-Klein, A., Wood, D.: One-Unambiguous Regular Languages. Information and Computation **140**(2), 229–253 (1998). `https://doi.org/10.1006/INCO.1997.2688`
20. Carapelle, C., Turhan, A.: Description Logics Reasoning w.r.t. General TBoxes Is Decidable for Concrete Domains with the EHD-Property. In: Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI). Frontiers in Artificial Intelligence and Applications, pp. 1440–1448 (2016). `https://doi.org/10.3233/978-1-61499-672-9-1440`
21. Chandra, A.K., Merlin, P.M.: Optimal Implementation of Conjunctive Queries in Relational Data Bases. In: Proceedings of the 9th Annual ACM Symposium on Theory of Computing (STOC), pp. 77–90 (1977). `https://doi.org/10.1145/800105.803397`

22. Chekuri, C., Rajaraman, A.: Conjunctive Query Containment Revisited. In: Proceedings of the 6th International Conference on Database Theory (ICDT). LNCS, vol. 1186, pp. 56–70. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-62222-5_36
23. Cocke, J.: Programming languages and their compilers: Preliminary notes, USA (1969)
24. Greiner, G., Hormann, K.: Efficient Clipping of Arbitrary Polygons. ACM Transactions on Graphics **17**(2), 71–83 (1998). https://doi.org/10.1145/274363.274364
25. Gruber, H., Holzer, M.: Finite Automata, Digraph Connectivity, and Regular Expression Size. In: Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP). LNCS, vol. 5126, pp. 39–50. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-70583-3_4
26. Haarslev, V., Lutz, C., Möller, R.: A Description Logic with Concrete Domains and a Role-forming Predicate Operator. Journal of Logic and Computation **9**(3), 351–384 (1999). https://doi.org/10.1093/LOGCOM/9.3.351
27. Haarslev, V., Möller, R., Wessel, M.: The Description Logic $\mathcal{ALCNH}_{R^+}$ Extended with Concrete Domains: A Practically Motivated Approach. In: Proceedings of the 1st International Joint Conference on Automated Reasoning (IJCAR). LNCS, vol. 2083, pp. 29–44. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45744-5_4
28. Hales, T.C.: The Jordan Curve Theorem, Formally and Informally. The American Mathematical Monthly **114**(10), 882–894 (2007). http://www.jstor.org/stable/27642361
29. Henzinger, T.A., Raskin, J.: The equivalence problem for finite automata: technical perspective. Communications of the ACM **58**(2), 86 (2015). https://doi.org/10.1145/2701001
30. Hickey, T.J., Ju, Q., van Emden, M.H.: Interval arithmetic: From principles to implementation. **48**(5), 1038–1068 (2001). https://doi.org/10.1145/502102.502106
31. Hopcroft, J.E., Karp, R.M.: A Linear Algorithm for Testing Equivalence of Finite Automata. Tech. rep. TR71-114, Cornell University (1971). https://hdl.handle.net/1813/5958
32. Hovland, D.: The inclusion problem for regular expressions. Journal of Computer and System Sciences **78**(6), 1795–1813 (2012). https://doi.org/10.1016/J.JCSS.2011.12.003
33. Jiang, S., Song, Z., Weinstein, O., Zhang, H.: A faster algorithm for solving general LPs. In: Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC), pp. 823–832 (2021). https://doi.org/10.1145/3406325.3451058
34. Jones, N.D.: Corrigendum: Space-Bounded Reducibility among Combinatorial Problems. Journal of Computer and System Sciences **15**(2), 241 (1977). https://doi.org/10.1016/S0022-0000(77)80009-3

35. Jones, N.D.: Space-Bounded Reducibility among Combinatorial Problems. Journal of Computer and System Sciences **11**(1), 68–85 (1975). See also [34]. `https://doi.org/10.1016/S0022-0000(75)80050-X`
36. Jordan, C.: Cours d'analyse de l'École Polytechnique — Volume 3: Calcul intégral, équations différentielles, Paris (1887)
37. Karakostas, G., Lipton, R.J., Viglas, A.: On the complexity of intersecting finite state automata and NL versus NP. Theoretical Computer Science **302**(1-3), 257–274 (2003). `https://doi.org/10.1016/S0304-3975(02)00830-7`
38. Karp, R.M.: Reducibility Among Combinatorial Problems. In: Proceedings of a Symposium on the Complexity of Computer Computations, held at the IBM Thomas J. Watson Research Center. The IBM Research Symposia Series, pp. 85–103 (1972). `https://doi.org/10.1007/978-1-4684-2001-2_9`
39. Kasami, T.: An Efficient Recognition and Syntax-Analysis Algorithm for Context-Free Languages. Report R-257, Coordinated Science Laboratory, University of Illinois, Urbana, Illinois (1966). `http://hdl.handle.net/2142/74304`
40. Kazakov, Y., Klinov, P.: Advancing ELK: Not Only Performance Matters. In: Proceedings of the 28th International Workshop on Description Logics (DL). CEUR Workshop Proceedings (2015). `https://ceur-ws.org/Vol-1350/paper-27.pdf`
41. Kazakov, Y., Krötzsch, M., Simančík, F.: Practical Reasoning with Nominals in the $\mathcal{EL}$ Family of Description Logics. In: Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR) (2012). `http://www.aaai.org/ocs/index.php/KR/KR12/paper/view/4540`
42. Kazakov, Y., Krötzsch, M., Simančík, F.: The Incredible ELK - From Polynomial Procedures to Efficient Reasoning with $\mathcal{EL}$ Ontologies. Journal of Automated Reasoning **53**(1), 1–61 (2014). `https://doi.org/10.1007/S10817-013-9296-3`
43. Kozen, D.: Lower Bounds for Natural Proof Systems. In: Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS), pp. 254–266 (1977). `https://doi.org/10.1109/SFCS.1977.16`
44. Kriegel, F.: Beyond Optimal: Interactive Identification of Better-than-optimal Repairs. In: Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing (SAC), pp. 1019–1026 (2025). `https://doi.org/10.1145/3672608.3707750`
45. Kriegel, F.: Optimal Fixed-Premise Repairs of $\mathcal{EL}$ TBoxes. In: Proceedings of the 45th German Conference on Artificial Intelligence (KI). LNCS, vol. 13404, pp. 115–130. Springer, Heidelberg (2022). `https://doi.org/10.1007/978-3-031-15791-2_11`
46. Kriegel, F.: Reasoning in OWL 2 EL with Hierarchical Concrete Domains (Extended Version). LTCS-Report 25-04, Chair of Automata Theory, In-

stitute of Theoretical Computer Science, Technische Universität Dresden (2025). https://doi.org/10.25368/2025.127

47. Krötzsch, M., Rudolph, S., Hitzler, P.: Complexities of Horn Description Logics. ACM Transactions on Computational Logic **14**(1), 2:1–2:36 (2013). https://doi.org/10.1145/2422085.2422087
48. Losemann, K., Martens, W., Niewerth, M.: Closure properties and descriptional complexity of deterministic regular expressions. Theoretical Computer Science **627**, 54–70 (2016). https://doi.org/10.1016/J.TCS.2016.02.027
49. Lutz, C.: NEXPTIME-complete description logics with concrete domains. ACM Transactions on Computational Logic **5**(4), 669–705 (2004). https://doi.org/10.1145/1024922.1024925
50. Lutz, C.: The complexity of description logics with concrete domains. Doctoral Thesis, RWTH Aachen University, Germany (2002). http://nbn-resolving.org/urn:nbn:de:hbz:82-opus-3032.
51. Lutz, C., Areces, C., Horrocks, I., Sattler, U.: Keys, Nominals, and Concrete Domains. Journal of Artificial Intelligence Research **23**, 667–726 (2005). https://doi.org/10.1613/JAIR.1542
52. Lutz, C., Miličić, M.: A Tableau Algorithm for Description Logics with Concrete Domains and General TBoxes. Journal of Automated Reasoning **38**(1-3), 227–259 (2007). https://doi.org/10.1007/S10817-006-9049-7
53. Magka, D., Kazakov, Y., Horrocks, I.: Tractable Extensions of the Description Logic $\mathcal{EL}$ with Numerical Datatypes. Journal of Automated Reasoning **47**(4), 427–450 (2011). https://doi.org/10.1007/S10817-011-9235-0
54. Martínez, F., Ogáyar, C.J., Jiménez, J., Ruiz, A.J.R.: A simple algorithm for Boolean operations on polygons. Advances in Engineering Software **64**, 11–19 (2013). https://doi.org/10.1016/J.ADVENGSOFT.2013.04.004
55. McNulty, G.F.: Fragments of First Order Logic, I: Universal Horn Logic. Journal of Symbolic Logic **42**(2), 221–237 (1977). https://doi.org/10.2307/2272123
56. O'Rourke, J., Chien, C., Olson, T., Naddor, D.: A new linear algorithm for intersecting convex polygons. Computer Graphics and Image Processing **19**(4), 384–391 (1982). https://doi.org/10.1016/0146-664X(82)90023-5
57. Ortiz, M., Rudolph, S., Šimkus, M.: Worst-Case Optimal Reasoning for the Horn-DL Fragments of OWL 1 and 2. In: Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning (KR) (2010). http://aaai.org/ocs/index.php/KR/KR2010/paper/view/1296
58. Pan, J.Z., Horrocks, I.: Reasoning in the $\mathcal{SHOQ}(\mathbf{D_n})$ Description Logic. In: Proceedings of the 15th International Workshop on Description Logics (DL). CEUR Workshop Proceedings (2002). https://ceur-ws.org/Vol-53/Pan-Horrocks-shoqdn-2002.ps
59. Rydval, J.: Using Model Theory to Find Decidable and Tractable Description Logics with Concrete Domains. Doctoral Thesis, Dresden University of

Technology, Germany (2022). https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa2-799074.

60. Shamos, M.I.: Computational Geometry. PhD thesis, Yale University, United States (1978). http://euro.ecom.cmu.edu/people/faculty/mshamos/1978ShamosThesis.pdf.
61. Stockmeyer, L.J., Meyer, A.R.: Word Problems Requiring Exponential Time: Preliminary Report. In: Proceedings of the 5th Annual ACM Symposium on Theory of Computing (STOC), pp. 1–9 (1973). https://doi.org/10.1145/800125.804029
62. Toussaint, G.T.: A simple linear algorithm for intersecting convex polygons. The Visual Computer **1**(2), 118–123 (1985). https://doi.org/10.1007/BF01898355
63. Vatti, B.R.: A Generic Solution to Polygon Clipping. Communications of the ACM **35**(7), 56–63 (1992). https://doi.org/10.1145/129902.129906
64. Younger, D.H.: Recognition and Parsing of Context-Free Languages in Time n^3. Information and Control **10**(2), 189–208 (1967). https://doi.org/10.1016/S0019-9958(67)80007-X
65. Yu, S., Zhuang, Q., Salomaa, K.: The State Complexities of Some Basic Operations on Regular Languages. Theoretical Computer Science **125**(2), 315–328 (1994). https://doi.org/10.1016/0304-3975(92)00011-F