SHACL Satisfiability: What Can We Learn from DLs?

Anouk Michelle Oudshoorn[1]

[1]Institute of Logic and Computation, TU Wien, Austria

Abstract

Since the introduction of the SHACL standard, understanding its computational features and formal foundations has become essential. Some research has focused on the semantics of recursive constraints and the complexity of validation, but the satisfiability of SHACL constraints remains largely unexplored. The most significant previous work in this direction is rather coarse, obtaining very few positive results for finite satisfiability and for fragments with counting. In this paper, we build on description logics to paint a comprehensive and fine-grained boundary for SHACL fragments with a decidable satisfiability problem under the supported semantics, both for unrestricted and finite models.

Keywords

SHACL, satisfiability, finite-model property

1. Introduction

Since the SHACL standard was introduced, the need for a solid understanding of its computational features and formal foundations has been apparent. Several works have leveraged related logic formalisms to give semantics to recursive constraints, obtain complexity bounds, and solve basic tasks including validation [1, 2, 3, 4], but little attention has been devoted to the satisfiability of SHACL constraints. This problem is of major importance in the design and validation of SHACL-based solutions: as SHACL becomes more popular, substantive efforts are put into its adoption. As part of this, we witness mining SHACL specifications from data [5, 6, 7], but how to assess the quality of these machine-generated constraints? And how to combine multiple, possibly generated, specifications? We note that the basic necessary condition here is compatibility, which boils down to satisfiability. A natural next step in assessing quality of data is tackling containment, for which satisfiability is a prerequisite. This, we plan to study in further work. Finally, both satisfiability and containment, as statistic analysis tools, are prerequisites for more advanced services like optimisation, incremental validation and modularity.

Given the importance of the problem, there are remarkably few results concerning its decidability and complexity. Indeed, the most notable work in this direction, [8], is very coarse. It builds on a tailored fragment of predicate logic to identify decidability and complexity bounds, but the basic logic it considers is already close to the boundary of what could potentially be decidable in the presence of cardinality constraints. The positive results are mostly limited to formalisms that do not support counting, and more often than not consider unrestricted (that is, potentially infinite) graphs, even though finite graphs are a more relevant setting here.

In this paper, we revisit satisfiability under the supported model semantics. We build on Description Logics (DLs), a well-known family of languages for Knowledge Representation and Reasoning that offers decades of research in the fine-grained study of logical fragments and the effect that the interaction between different shapes of subformulas has on the complexity of reasoning. The close relationship between DLs and SHACL is well-known, and in this paper, we leverage it to paint a much finer boundary of SHACL fragments that have decidable satisfiability problems, both over unrestricted graphs and over graphs with a finite domain.

DL 2025: 38th International Workshop on Description Logics, September 3–6, 2025, Opole, Poland

anouk.oudshoorn@tuwien.ac.at (A. M. Oudshoorn)
0009-0006-4638-5948 (A. M. Oudshoorn)

Contributions. We build on the DL literature to pinpoint much tighter complexity bounds than previously known for SHACL, based on the close connection between DL - and SHACL satisfiability; we revisit this connection and explain how to translate complexity results in both ways. To emphasise this tight bond, we provide a DL inspired naming convention: we write $\mathcal{L}_S$ to denote the SHACL fragment similar to the DL $\mathcal{L}$. Moreover, we add some lack of finite model property results to the landscape: we show this for $\mathcal{ALC}_S$ plus counting over regular path expressions, which also provides an alternative undecidability proof; and, we show that adding either $\mathsf{eq}(E, r)$ or $\mathsf{disj}(E, r)$ to $\mathcal{ALC}_S$ also breaks the finite model property of $\mathcal{ALC}_S$.

Related Literature. There are two other theoretical papers considering satisfiability of (recursive) SHACL [8, 9]. Both works are based on a translation of SHACL into a fragment of first-order logic and transferring complexity results. A tool for testing SHACL satisfiability based on this translation is presented in [10]. Our work differs in considering different fragments by starting from a smaller base logic: the smallest logic considered in those works corresponds to $\mathcal{ALCIO}_S$ extended with universal roles. Another work considering the close connection between SHACL and DLs for deciding complexity of reasoning problems, in their case shape containment, is [11]. However, as pointed out in [12], there are some issues with their translation.

2. Preliminaries

Data Graphs and Interpretations. Let N_C, N_R and N_I denote countably infinite, mutually disjoint sets of *concept names*, *role names*, and *individuals*, respectively. Let $N_R^+ := \{p, p^- \mid p \in N_R\}$ be the set of *roles*. For every $p \in N_R$, set $(p^-)^- = p$. An *atom* is an expression of the form $A(c)$ or $p(c, c')$, for $A \in N_C$, $p \in N_R$ and $\{c, c'\} \subseteq N_I$. An *ABox* (or *data graph*) $\mathcal{A}$ is a finite set of atoms.

An *interpretation* is a pair $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty set (called *domain*) and $\cdot^\mathcal{I}$ is a function that maps every $A \in N_C$ to a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, every $p \in N_R$ to a binary relation $p^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, and every individual $c \in N_I$ to an element $c^\mathcal{I} \in \Delta^\mathcal{I}$. Let $(p^-)^\mathcal{I} := \{(c', c) \mid (c, c') \in p^\mathcal{I}\}$. We call an interpretation $\mathcal{I}$ *finite* when $\Delta^\mathcal{I}$ is finite. We make the *standard name assumption*, meaning $c^\mathcal{I} = c$ for all interpretations $\mathcal{I}$, and all $c \in N_I$. The *canonical interpretation* $\mathcal{I}_\mathcal{A}$ of a set of atoms $\mathcal{A}$ is defined by setting $\Delta^{\mathcal{I}_\mathcal{A}} = \{c \mid A(c) \in \mathcal{A}\} \cup \{(c, c') \mid p(c, c') \in \mathcal{A}\}$, $A^{\mathcal{I}_\mathcal{A}} = \{c \mid A(c) \in \mathcal{A}\}$ for all $A \in N_C$ and $p^{\mathcal{I}_\mathcal{A}} = \{(c, c') \mid p(c, c') \in \mathcal{A}\}$ for all $p \in N_R$.

Description Logic $\mathcal{ALCOIQ}$. An $\mathcal{ALCOIQ}$ concept C is defined in the following way:

$$C ::= c \mid A \mid \top \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \geq_n r.C \mid \forall r.C,$$

where $c \in N_I$, $A \in N_C$, $n \geq 1$ and $r \in N_R^+$. An $\mathcal{ALCOIQ}$ TBox $\mathcal{T}$ is a set of axioms of the form $C \sqsubseteq D$, for C and D $\mathcal{ALCOIQ}$ concepts. We use $C \equiv D$ as a shorthand for $C \sqsubseteq D$ and $D \sqsubseteq C$. An interpretation $\mathcal{I}$ is a model of $\mathcal{T}$ if for all $C \sqsubseteq D \in \mathcal{T}$ we have $C^\mathcal{I} \subseteq D^\mathcal{I}$, where $C^\mathcal{I}$ is recursively defined as: $(\neg C)^\mathcal{I} := \Delta^\mathcal{I} \setminus C^\mathcal{I}$, $(C \sqcap C')^\mathcal{I} := C^\mathcal{I} \cap C'^\mathcal{I}$, $(C \sqcup C')^\mathcal{I} := C^\mathcal{I} \cup C'^\mathcal{I}$, $(\geq_n r.C)^\mathcal{I} := \{c \in \Delta^\mathcal{I} \mid |\{c' \in \Delta^\mathcal{I} \mid (c, c') \in r^\mathcal{I}, c' \in C^\mathcal{I}\}| \geq n\}$ and $(\forall r.C)^\mathcal{I} := \{c \in \Delta^\mathcal{I} \mid (c, c') \in r^\mathcal{I} \to c' \in C^\mathcal{I}\}$. A concept C is *satisfiable* w.r.t. a TBox $\mathcal{T}$ if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^\mathcal{I} \neq \emptyset$.

Recursive Shape Constraint Language (SHACL). Let N_S be a countably infinite set of shape names, disjoint from N_I, N_R and N_C. We define *shape expressions*, following [13], but adding recursion, in the following way

$$\varphi ::= s \mid c \mid A \mid \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid \geq_n E.\varphi \mid \mathsf{eq}(E, r) \mid \mathsf{disj}(E, r) \mid \mathsf{closed}(R),$$

where $s \in N_S$, $c \in N_I$, $A \in N_C$, $n \geq 1$, R a finite subset of N_R^+ and E a regular expression given by

$$E ::= r \mid E^* \mid E \circ E \mid E \cup E,$$

$$\begin{aligned}
\top^{\mathcal{I},S} &= N_I & s^{\mathcal{I},S} &= \{c \in \Delta^{\mathcal{I}} \mid s(c) \in S\} \\
c^{\mathcal{I},S} &= \{c^{\mathcal{I}}\} & (\neg\varphi)^{\mathcal{I},S} &= \Delta^{\mathcal{I}} \setminus (\varphi)^{\mathcal{I},S} \\
A^{\mathcal{I},S} &= A^{\mathcal{I}} & (\varphi \wedge \varphi')^{\mathcal{I},S} &= (\varphi)^{\mathcal{I},S} \cap (\varphi')^{\mathcal{I},S}
\end{aligned}$$

$$\begin{aligned}
(\geq_n E.\varphi)^{\mathcal{I},S} &= \{c \in \Delta^{\mathcal{I}} \mid |\{c' \in \Delta^{\mathcal{I}} \mid (c,c') \in E^{\mathcal{I}}, c' \in \varphi^{\mathcal{I},S}\}| \geq n\} \\
(\mathsf{eq}(E,r))^{\mathcal{I},S} &= \{c \in \Delta^{\mathcal{I}} \mid \{c' \in \Delta^{\mathcal{I}} \mid (c,c') \in E^{\mathcal{I}}\} = \{c' \in \Delta^{\mathcal{I}} \mid (c,c') \in r^{\mathcal{I}}\}\} \\
(\mathsf{disj}(E,r))^{\mathcal{I},S} &= \{c \in \Delta^{\mathcal{I}} \mid \{c' \in \Delta^{\mathcal{I}} \mid (c,c') \in E^{\mathcal{I}}\} \cap \{c' \in \Delta^{\mathcal{I}} \mid (c,c') \in r^{\mathcal{I}}\} = \emptyset\} \\
(\mathsf{closed}(R))^{\mathcal{I},S} &= \{c \in \Delta^{\mathcal{I}} \mid \{r \in N_R^+ \setminus R \mid (c,c') \in r^{\mathcal{I}}\} = \emptyset\}
\end{aligned}$$

Figure 1: Evaluating shape expressions

for $r \in N_R^+$. Here, $(E^*)^{\mathcal{I}}$ corresponds to the transitive closure of $E^{\mathcal{I}}$, $(E \circ E')^{\mathcal{I}} := \{(c,c') \mid (c,d) \in E^{\mathcal{I}}, (d,c') \in E'^{\mathcal{I}}\}$, and $(E \cup E')^{\mathcal{I}} := E^{\mathcal{I}} \cup E'^{\mathcal{I}}$. We use EE' as a shorthand for $E \circ E'$, and E^+ for EE^*. We set $\varphi \vee \varphi' := \neg(\neg\varphi \wedge \neg\varphi')$ and $\forall E.\varphi := \neg \geq_1 E.\neg\varphi$. A *shape constraint* is an expression of the form $s \leftarrow \varphi$, for $s \in N_S$ and φ a shape expression. With $\mathcal{C}$, we indicate a set of shape constraints. For each $s \leftarrow \varphi$, let s be the *head* of the constraint. In each $\mathcal{C}$, we assume each shape name s only appears as the head of one constraint - this does not influence expressivity as '$\vee$' may be used.

A shape atom is an expression of the form $s(c)$, for $s \in N_S$ and $c \in N_I$. A *shape assignment* S is a set of shape atoms. Given an interpretation $\mathcal{I}$ and a shape assignment S, we say a individual $c \in N_I$ *validates* a shape expression φ, whenever $c \in (\varphi)^{\mathcal{I},S}$, where $(\varphi)^{\mathcal{I},S}$ is recursively defined in Table 1. Given some $\mathcal{C}$, we say c validates $s \in N_S$, if c validates φ for all $s \leftarrow \varphi \in \mathcal{C}$. Let $\mathcal{G}$ be a set of targets of the form $s(c)$, which we call *atomic targets*, or $s(A)$, for $s \in N_S$, $c \in N_I$ and $A \in N_C$. A pair $(\mathcal{C}, \mathcal{G})$ is called a *shapes graph*. In this paper, we consider the *supported model semantics*; given an interpretation $\mathcal{I}$, we say $\mathcal{I}$ *validates* $(\mathcal{C}, \mathcal{G})$ when there exists a shape assignment S such that if $s \leftarrow \varphi \in \mathcal{C}$, we find $s^{\mathcal{I},\mathcal{C}} = (\varphi)^{\mathcal{I},\mathcal{C}}$ and for all $s(c) \in \mathcal{G}$, we find c validates s, and for all $s(A) \in \mathcal{G}$, all individuals in $\mathcal{A}^{\mathcal{I}}$ validate s. Different semantics require different constraints for the shape assignments. For readability, we will write $\mathcal{A}$ validates $(\mathcal{C}, \mathcal{G})$, for a set of atoms $\mathcal{A}$ to mean that the canonical interpretation $\mathcal{I}_{\mathcal{A}}$ validates $(\mathcal{C}, \mathcal{G})$.

3. SHACL Satisfiability

In this paper we study the following reasoning problems:

Satisfiability: Given a SHACL fragment $\mathcal{L}_S$, for each shapes graph $(\mathcal{C}, \mathcal{G})$ expressible in $\mathcal{L}_S$, decide whether there exists an interpretation $\mathcal{I}$ that validates $(\mathcal{C}, \mathcal{G})$.

Finite Satisfiability: Given a SHACL fragment $\mathcal{L}_S$, for each shapes graph $(\mathcal{C}, \mathcal{G})$ expressible in $\mathcal{L}_S$, decide whether there exists a *finite* interpretation $\mathcal{I}$ that validates $(\mathcal{C}, \mathcal{G})$.

We also study the following property, which guarantees that these problems coincide:

Finite Model Property: A SHACL fragment $\mathcal{L}_S$ has the finite model property iff for every shapes graph $(\mathcal{C}, \mathcal{G})$ expressible in $\mathcal{L}_S$, we find that if $(\mathcal{C}, \mathcal{G})$ is satisfiable, then $(\mathcal{C}, \mathcal{G})$ is finitely satisfiable.

Clearly, having the finite model property extends to less expressive fragments, whereas the opposite, not having the property, spreads to subsuming fragments. Similarly, for (finite) satisfiability, membership of a complexity class spreads to less expressive fragments, and hardness to the more expressive ones. In case a fragment has the finite model property, the membership results for general satisfiability extend to the finite setting.

The above presented problems are not the only ones one might consider: in [9], also another flavour of the SHACL satisfiability problem is discussed: *constraint satisfiability*. This corresponds to the satisfiability problem when the constraint set only consists of one constraint, and with no extra restrictions on the target set $\mathcal{G}$. As already noted in [9], the constraint version of the problem clearly reduces to the general version, which means upper bounds for complexity are preserved. We show here that for recursive SHACL also the other reduction holds. First, we note that for satisfiability purposes, we may restrict the form of the targets.

Lemma 1. *For each shapes graph $(\mathcal{C}, \mathcal{G})$ there exists a shapes graph $(\mathcal{C}', \mathcal{G}')$ such that $\mathcal{G}'$ only consists of atomic targets and for each model $\mathcal{I}$ we have $\mathcal{I}$ validates $(\mathcal{C}, \mathcal{G})$ iff $\mathcal{I}$ validates $(\mathcal{C}', \mathcal{G}')$.*

Proof. Assume that for some concept name $A \in N_C$, $s(A) \in \mathcal{G}$. It suffices to replace each occurrence of A in $\mathcal{C}$ by $(A \wedge s)$, and remove $s(A)$ from $\mathcal{G}$. In this way, we enforce that each node with an A-label, essential in the validation of another constraint, also validates s. □

Proposition 1. *In recursive SHACL, the problems of deciding SHACL satisfiability and constraint satisfiability are mutually reducible.*

Proof. We use '$\varphi \leftrightarrow \psi$' as a shorthand for $(\varphi \to \psi) \wedge (\psi \to \varphi)$, and '$\varphi \to \psi$' for $\neg\varphi \vee \psi$. Given a shapes graph $(\mathcal{C}, \mathcal{G})$, such that all targets in $\mathcal{G}$ are atomic. We distinguish two cases.

In case the considered SHACL fragment does not contain nominals, satisfiability of $(\mathcal{C}, \mathcal{G})$ is equivalent to satisfiability of all $(\mathcal{C}, \mathcal{G}_c)$ separately, where $\mathcal{G}_c := \{s(c) \in \mathcal{G}\}$, for all $c \in N_I$ such that c appears in $\mathcal{G}$. Furthermore, note we may replace multiple targets using the same c by a single target $s(c)$ for some fresh shape name s, given we add $s \leftarrow \bigwedge_{s'(c) \in \mathcal{G}} s'$ to the set of constraints. Thus, we further assume that $\mathcal{G} = \{s(c)\}$.

The next step is to encode all constraints within a single one: satisfiability of $(\mathcal{C}, \{s(c)\})$ can be reduced to satisfiability of $(\{\hat{s} \leftarrow \hat{\varphi}\}, \{\hat{s}(c)\})$, for a fresh shape name $\hat{s}$, and $\hat{\varphi}$ defined in the following way:

$$\hat{\varphi} := s \wedge \forall(\bigsqcap_{r \in R} r)^*. \bigwedge_{s' \leftarrow \varphi \in \mathcal{C}} (s' \leftrightarrow \varphi),$$

where $R \subseteq N_R^+$ contains all roles appearing in any constraint in $\mathcal{C}$.

In the case the SHACL fragment does contain nominals, the above described reduction to single-element targets may no longer be sound. Instead, we use the nominals in the newly defined constraint in the following way: satisfiability of $(\mathcal{C}, \mathcal{G})$ may be reduced to satisfiability of $(\{\hat{s} \leftarrow \hat{\varphi}\}, \{\hat{s}(c)\})$ for each $c \in N_I$ appearing in $\mathcal{G}$, such that

$$\hat{\varphi} := \forall(\bigsqcap_{r \in R} r)^*. \bigwedge_{s(c) \in \mathcal{G}} (c \to s) \wedge \bigwedge_{s \leftarrow \varphi \in \mathcal{C}} (s \leftrightarrow \varphi),$$

where $R \subseteq N_R^+$ contains all roles appearing in any constraint in $\mathcal{C}$. □

Names for fragments of SHACL. Let $\mathcal{ALC}_S$ be the fragment of SHACL such that shape expressions φ are of the form:

$$\varphi ::= s \mid A \mid \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists r.\varphi \mid \forall r.\varphi,$$

for $r \in N_R$. Let $\exists r.\varphi$ be a shorthand for $\geq_1 r.\varphi$. Partly following the naming convention of Description Logics, we identify the SHACL fragments in the way presented in Table 1. We write $\mathcal{L}X_S$ to denote the SHACL fragment by extending $\mathcal{L}_S$ with the features described by some $X \subseteq \{\mathcal{O}, \mathcal{I}, \mathcal{F}, \mathcal{N}, \mathcal{Q}, \mathcal{E}, \mathcal{P}\}$. With the superscript $\mathcal{L}^e$, we denote that the feature $\mathsf{eq}(r, r')$, for $\{r, r'\} \subseteq N_R$ is added to the fragment $\mathcal{L}$. Similarly, $\mathcal{L}^d$ corresponds to adding the feature $\mathsf{disj}(r, r')$, also for $\{r, r'\} \subseteq N_R$. In case the fragment $\mathcal{L}$ contains the letter $\mathcal{I}$, $\{r, r'\} \subseteq N_R^+$

Note that adding $\mathsf{closed}(R)$ does not increase the expressivity of $\mathcal{ALC}_S$. Introducing $\geq_1 E.\varphi$ does increase expressivity of $\mathcal{ALC}_S$ in the supported model semantics, but not in, among others, the least-fixed point semantics [14].

Lemma 2. *For each shapes graph $(\mathcal{C}, \mathcal{G})$ expressible in $\mathcal{ALC}_S$ extended with expressions of the form* $\mathsf{closed}(R)$*, there exists a constraint set $\mathcal{C}'$ expressible in $\mathcal{ALC}_S$ such that $(\mathcal{C}, \mathcal{G})$ is (finitely) satisfiable iff $(\mathcal{C}', \mathcal{G})$ is (finitely) satisfiable.*

Proof. Since we are in the restricted context of SHACL satisfiability, that is, roles not mentioned in the constraints are irrelevant, we may replace each occurrence of '$\mathsf{closed}(R)$' by '$\bigsqcap_{r \in R^c} \neg\exists r.\top$', where $R^c := \{r \in N_R \setminus R \mid r \text{ appears in } \mathcal{C}\}$, to construct $\mathcal{C}'$. □

Name	Syntax	Symbol
Nominals	c	$\mathcal{O}$
Inverses	r^-	$\mathcal{I}$
Functionality	$\leq_1 r.\top$	$\mathcal{F}$
Unqualified number restriction	$\geq_n r.\top$	$\mathcal{N}$
Qualified number restriction	$\geq_n r.\varphi$	$\mathcal{Q}$
Unqualified regular path counting	$\geq_n E.\top$	$\mathcal{E}$
Qualified regular path counting	$\geq_n E.\varphi$	$\mathcal{P}$

Table 1
Fragments of SHACL following the DL naming convention, extended with counting over regular paths.

$$\begin{aligned}
f(c) &:= c & g(c) &:= c\\
f(A) &:= s_A & g(A) &:= A,\, g(s) := A_s\\
f(\top) &:= \top & g(\top) &:= \top\\
f(\neg C) &:= \neg f(C) & g(\neg\varphi) &:= \neg g(\varphi)\\
f(C \sqcap C') &:= f(C) \wedge f(C') & g(\varphi \wedge \varphi') &:= g(\varphi) \sqcap g(\varphi')\\
f(C \sqcup C') &:= f(C) \vee f(C') & g(\varphi \vee \varphi') &:= g(\varphi) \sqcup g(\varphi')\\
f(\geq_n r.C) &:= \geq_n r.f(C) & g(\geq_n r.\varphi) &:= \geq_n r.g(\varphi)\\
f(\forall r.C) &:= \forall r.f(C) & g(\forall r.\varphi) &:= \forall r.g(\varphi)
\end{aligned}$$

Figure 2: Translation functions mapping $\mathcal{ALCOIQ}$ concepts into $\mathcal{ALCOIQ}_S$ shape expressions, and vice versa.

4. SHACL to OWL and back again

Most of the results in this work are based on the tight connection between SHACL and DLs. In this section, we look at their connection and provide a translation for satisfiability purposes.

Translation. We note that for $\mathcal{ALC}$ and more expressive DLs, it is immediate that we can restrict the logic to equivalence axioms only, without affecting its expressivity. That is, $C \sqsubseteq D$ may be replaced by $\top \equiv \neg C \sqcup D$. In these cases, we may also assume without loss of generality that one side of the equivalence is a concept name: it is always possible to introduce a fresh concept name as middle ground. Furthermore, we note that when considering satisfiability of a concept name A w.r.t. a TBox $\mathcal{T}$, we can reduce axioms of the form $\top \equiv C$ to $A_C \equiv \bigsqcup_{r \in R} \forall r.(A_C \sqcap C)$, where $R \subseteq N_R^+$ contains all roles appearing in $\mathcal{T}$. In this case, we find that A is satisfiable w.r.t. $\mathcal{T}$ iff $A \sqcap A_C$ is satisfiable w.r.t. $(\mathcal{T} \cup \{A_C \equiv \bigsqcup_{r \in R} \forall r.(A_C \sqcap C)\}) \setminus \{\top \equiv C\}$. That is, in this paper, it will be sufficient to consider axioms of the form $A \equiv C$, for $A \in N_C \setminus \top$. Moreover, we assume that each considered TBox $\mathcal{T}$ contains for each $A \in N_C$ at most one concept C, possibly making use of '$\sqcup$', such that $A \equiv C \in \mathcal{T}$. As we set $s \leftarrow \varphi \in \mathcal{C}$ implies $(s)^{\mathcal{I},S} = (\varphi)^{\mathcal{I},S}$, this aligns well with the semantics of recursive SHACL we are considering.

Let us define two translations: f, a function translating any $\mathcal{ALCOIQ}$ concept into a shape expression, and g, a function in the opposite direction, translating any shape expression expressible in $\mathcal{ALCOIQ}_S$ into an $\mathcal{ALCOIQ}$ concept. These functions are recursively defined in Table 2, where $s_A \in N_S$ is a fresh shape name introduced for every concept name $A \in N_C$, and A_s is a fresh concept name for each $s \in N_S$. Note that fragments are preserved: an $\mathcal{L}_S$ shape expression translates into an $\mathcal{L}$ concept, and vice versa.

Proposition 2. *Let $\mathcal{T}$ be an $\mathcal{ALCOIQ}$ TBox such that all axioms are of the form $A \equiv C$, and such that no pair $\{B \equiv C, B \equiv C'\}$, for $C \neq C'$ is contained in $\mathcal{T}$. Then, $\mathcal{I}$ is a model of $\mathcal{T}$ such that $A^{\mathcal{I}} \neq \emptyset$ iff $\mathcal{I}$ validates $(\mathcal{C}, \mathcal{G})$ given by $\mathcal{G} = \{s_A(c) \mid c \in A^{\mathcal{I}}\}$ and*

$$\mathcal{C} = \{s_A \leftarrow f(C) \wedge A \mid A \equiv C \in \mathcal{T}\} \cup \{s_A \leftarrow A \mid A \equiv C \notin \mathcal{T}\}.$$

Proposition 3. *Let $(\mathcal{C}, \mathcal{G})$ be any $\mathcal{ALCOIQ}_S$ shapes graph such that $\mathcal{G}$ only contains atomic targets. Then $\mathcal{I}$ validates $(\mathcal{C}, \mathcal{G})$, because of the shape assignment S, iff $\mathcal{I}'$ is a model of $\{A_s \equiv g(\varphi) \mid s \leftarrow \varphi \in \mathcal{C}\}$,* *such that if $s(c) \in \mathcal{G}$, then $c \in A_s^{\mathcal{I}'}$. Here, $\mathcal{I}'$ has domain $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}'}$, and is further defined as: for all $A \in N_C \setminus \{A_s \mid s \in N_S\}$, $A^{\mathcal{I}} = A^{\mathcal{I}'}$ and for all $A \in \{A_s \mid s \in N_S\}$, $A_s^{\mathcal{I}'} = \{c \in N_I \mid s(c) \in S\}$.*

Note that correctness of both propositions is based on the fact that shape and concept names can be considered as very similar, namely as unary labels for individuals, in the setting of determining (finite) satisfiability.

Joint Satisfiability of SHACL and OWL. As envisioned in the W3C SHACL specification [15, Section 1.5] and argued in [3], it is promising to combine SHACL and OWL (under the unique name assumption), another prominent W3C standard for managing data, whose profiles are based on DLs [16]. Combining these formalisms gives rise to a whole new set of challenges, like how to reconcile the open- and closed-world semantics these specifications bring along [3, 17]. Fortunately, the semantics proposed in [3] and [14], i.e., SHACL validation over the core universal model of the A- and TBox, can be reduced to plain SHACL validation [3, 14]. Also the complexity of validation is discussed there. However, nothing is known regarding joint satisfiability of SHACL and OWL, that is, the following reasoning problems.

Joint Satisfiability: Given a SHACL fragment $\mathcal{L}_S$ and OWL fragment $\mathcal{L}'$, for each shapes graph $(\mathcal{C}, \mathcal{G})$ expressible in $\mathcal{L}_S$ and each TBox $\mathcal{T}$ expressible in $\mathcal{L}'$, decide whether there exists an interpretation $\mathcal{I}$ that validates $(\mathcal{C}, \mathcal{G})$ and is a model of $\mathcal{T}$.

Finite Joint Satisfiability: Given a SHACL fragment $\mathcal{L}_S$ and OWL fragment $\mathcal{L}'$, for each shapes graph $(\mathcal{C}, \mathcal{G})$ expressible in $\mathcal{L}_S$ and each TBox $\mathcal{T}$ expressible in $\mathcal{L}'$, decide whether there exists a finite interpretation $\mathcal{I}$ that validates $(\mathcal{C}, \mathcal{G})$ and is a model of $\mathcal{T}$.

Given the above presented translation, it follows that the complexity of deciding (finite) joint satisfiability of SHACL in presence of OWL corresponds to the complexity of deciding (finite) satisfiability in the least-expressive description logic capturing the expressivity of both the translated SHACL fragment, as the OWL fragment.

5. Inverses, Nominals and Counting

The following propositions are well-known results in the Description Logic community. These results extend to our setting, using a translation as described in the previous section.

Proposition 4 (for instance [18, 19, 20])**.** *$\mathcal{ALCOI}_S$ and $\mathcal{ALCOQ}_S$ have the finite model property, $\mathcal{ALCIF}_S$ does not.*

Proposition 5 ([20, 21], and their references)**.** *Deciding (finite) satisfiability in $\mathcal{ALC}_S$, $\mathcal{ALCOI}_S$, $\mathcal{ALCOQ}_S$, and $\mathcal{ALCIQ}_S$ is ExpTime-complete.*

Proposition 6. *Deciding (finite) satisfiability in $\mathcal{ALCOIF}_S$ and $\mathcal{ALCOIQ}_S$ is NExpTime-complete.*

The lower bound for $\mathcal{ALCOIF}_S$ follows from constructing a torus of finite size [22]; the upper bound from translating $\mathcal{ALCOIQ}_S$ into the two-variable fragment of first-order logic with counting quantifiers C^2 [23], in which (finite) satisfiability is NExpTime-complete [24].

Proposition 7. *$\mathcal{ALCE}_S$ and $\mathcal{ALCP}_S$ do not have the finite model property.*

Proof. Consider the following constraints, with the target $s(0, 0)$:

$$s \leftarrow \forall u^+ . =_1 r^+ d.\top \land \forall (r \cup u)^* . (s_f \land s_g)$$
$$s_f \leftarrow =_1 u.\top \land =_1 r.\top \land =_1 (ru \cup ur).\top$$
$$s_g \leftarrow \neg \geq_1 d.\top \lor \forall u^+ . \neg \geq_1 d.\top$$

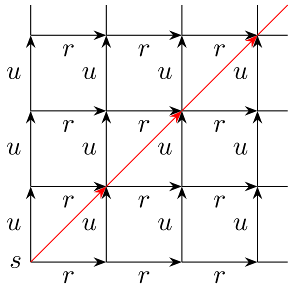

Figure 3: Infinite grid that, after adding s_f and s_g as label to every node, shows satisfiability of s. The red diagonal arrows denote the role d.

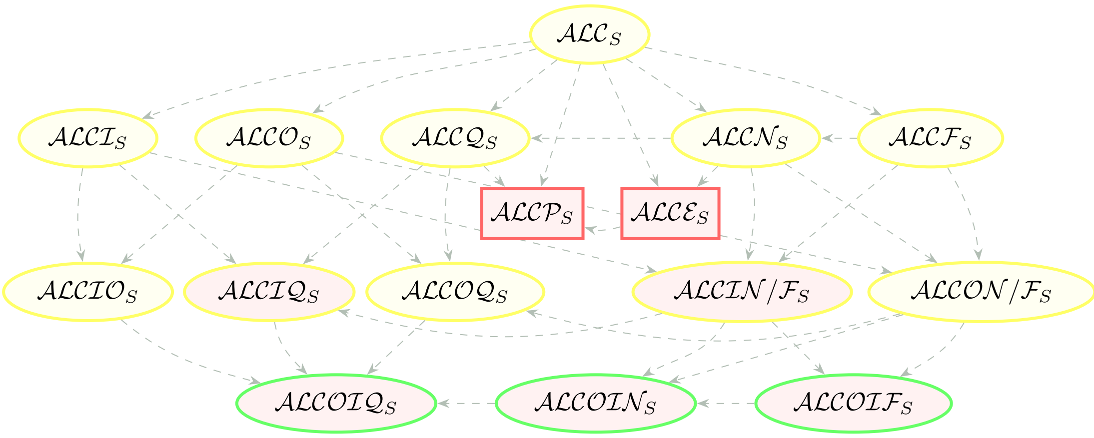

Figure 4: Decidability and complexity of SHACL fragments. Ellipse-shaped nodes denote (finite) satisfiability is decidable in ExpTime (yellow border), or NExpTime (green border). Squared-shaped nodes indicate satisfiability is undecidable. A yellow filling indicates the presence of the finite model property, whereas a red filling stands for the lack of it. Arrows indicate subsumption of fragments.

Here, $=_1 .\varphi$ is a shorthand for $\geq_1 .\varphi \wedge \leq_1 .\varphi$. Clearly, a way to satisfy the above constraints is in a simple grid on the natural numbers with a diagonal, where s true in $(0, 0)$ and s_f and s_g validated everywhere. Here the interpretation of d is $\{(i, i), (i + 1, i + 1) \mid i \in \mathbb{N}\}$, for u it is $\{((i, j), (i, j + 1)) \mid \{i, j\} \subseteq \mathbb{N}\}$, and for r the set $\{((i, j), (i + 1, j)) \mid \{i, j\} \subseteq \mathbb{N}\}$.

Assume for contradiction there exists a finite model. As s_f must hold in s and every individual reachable by u, there exists $a_0, \ldots, a_i$ such that a_0 is reachable by u^* from $(0, 0)$ and $\{(a_0, a_1), \ldots, (a_{i-1}, a_i), (a_i, a_0)\}$ is contained in the interpretation of u. Note that because of having to validate $=_1 r.\top \wedge =_1 (ru \cup ur).\top$ in every individual reachable by r or u, it can be concluded that the set of individuals $\{b_0, \ldots, b_j\}$ reachable by r from any individual in $\{a_0, \ldots, a_i\}$ must also contain a loop in the interpretation of u. Clearly, this generalises to: every individual reachable by r^+ from any individual in $\{a_0, \ldots, a_i\}$ has a u^+-path leading to itself. As every individual appearing in a loop of u's cannot have an outgoing d-edge, because of the constraint $s_g \leftarrow \neg{\geq_1} d.\top \vee \forall u^+.\neg{\geq_1} d.\top$, it follows that every individual reachable by r^+ from any individual in $\{a_0, \ldots, a_i\}$ cannot have an outgoing d-edge. As all individuals in $\{a_0, \ldots, a_i\}$ are reachable by u^+ from $(0, 0)$, we cannot validate the first conjunct of s in $(0, 0)$. This is the contradiction which concludes the proof. □

Note the above proof produces a grid, which means only a few more rules need to be introduced to reduce the undecidable domino problem [25] to $\mathcal{ALCE}_S$. It is easy to check this is possible, making the satisfiability problem undecidable. This result is already known for different sublogics of $\mathcal{ALCE}_S$, which is discussed in the remainder of this section.

More Fine-Grained Analysis. In the following, we will restrict the expressivity of the regular expressions used in $\geq_n E.\top$ and $\geq_n E.\varphi$. That is, with $\mathcal{ALCN}(X)_S$ or $\mathcal{ALCQ}(X)_S$, for X any combination of the role constructs $*, \circ$ and $\cup$, we denote the SHACL fragment allowing regular expressions build from only the role constructs in X in number restrictions. That is, $\mathcal{ALCN}(*, \circ, \cup)_S = \mathcal{ALCE}_S$ and $\mathcal{ALCQ}(*, \circ, \cup)_S = \mathcal{ALCP}_S$. We note that the translation presented in Section 4 naturally extends to also capture $*$, $\circ$ and $\cup$ in the number restrictions. Again, we can rely on the vast DL literature: the derived complexity results are the following.

Proposition 8. *Satisfiability in $\mathcal{ALCN}(\circ)_S$ is undecidable.*

This is a direct consequence of Theorem 6 in [26].

Proposition 9. *Satisfiability in $\mathcal{ALCN}(*, \cup)_S$ is undecidable.*

Proof. We can adapt the undecidability proof of unrestricted $\mathcal{SHN}$ in [27] in the following way. That is, instead of using the hierarchy and the given axioms, we consider the following shape expressions.

$$s_A \leftarrow \neg s_B \wedge \neg s_C \wedge \neg s_D \wedge \exists x_1.s_B \wedge \exists y_1.s_C \wedge \leq_3 (x_1 \cup y_1)^*.\top$$
$$s_B \leftarrow \neg s_A \wedge \neg s_C \wedge \neg s_D \wedge \exists x_2.s_A \wedge \exists y_1.s_D \wedge \leq_3 (x_2 \cup y_1)^*.\top$$
$$s_C \leftarrow \neg s_A \wedge \neg s_B \wedge \neg s_D \wedge \exists x_1.s_D \wedge \exists y_2.s_A \wedge \leq_3 (x_1 \cup y_2)^*.\top$$
$$s_D \leftarrow \neg s_A \wedge \neg s_B \wedge \neg s_C \wedge \exists x_2.s_C \wedge \exists y_2.s_B \wedge \leq_3 (x_2 \cup y_2)^*.\top$$

Note that satisfiability of $s_A(c)$ corresponds to existence of a grid. Now it is easy to check we can encode a domino tiling problem like in [28]. Thus, the undecidability of the domino problem transfers to this logic, which concludes our proof. □

Deciding (finite) satisfiability in $\mathcal{ALCQ}(\cup)_S$ is ExpTime-complete. This result is subsumed by Proposition 12 in the next section.

6. Equality and Disjointness

Recall we introduced the superscripts $\mathcal{L}^d$ and $\mathcal{L}^e$ to denote the addition of the features $\mathsf{disj}(r, r')$ and $\mathsf{eq}(r, r')$, respectively. Following the naming convention introduced in the previous section, for X any combination of the role constructs $*, \circ$ and $\cup$, let $\mathcal{L}(X)^d$, resp. $\mathcal{L}(X)^e$, be the SHACL fragment allowing regular expressions build from only the role constructs in X in the disjointness, resp. equality feature, and in number restrictions, in case $\mathcal{N}$ or $\mathcal{Q}$ is contained in $\mathcal{L}$. That is, recursive SHACL as introduced in the preliminaries, and for satisfiability purposes, corresponds to $\mathcal{ALCOIQ}(*, \circ, \cup)_S^{d,e}$.

We start with a positive result: adding disjointness does not increase complexity, although the finite model property is easily lost.

Proposition 10. *Deciding satisfiability in $\mathcal{ALCI}(*, \circ, \cup)_S^{\mathbf{d}}$ is ExpTime-complete, and this fragment does not have the finite model property. In fact, $\mathcal{ALC}(*, \circ)_S^{\mathbf{d}}$ already lacks this property.*

Proof. The upper bound follows from Theorem 4.8 in [29]. To see this, note that $\mathsf{disj}(E, r)$ is equivalent to the expression $\forall(E \cap r).\bot$. As the amount of nestings of '$\cap$' in this expression is bounded by a constant, namely 1, the tighter upper bound of ExpTime can be derived.

For the lack of finite model property, consider the following shapes graph $(\mathcal{C}, \mathcal{G})$:

$$\mathcal{C} = \{s \leftarrow \mathsf{disj}(rr^+, r) \wedge \exists r.s\}$$

and set $\mathcal{G} = \{s(a)\}$. Clearly, the infinite chain of r's, in which every individual is labelled with an s is an infinite model. In fact, it must be possible to homomorphically map this chain into any interpretation that validates $(\mathcal{C}, \mathcal{G})$. As $\mathsf{disj}(rr^+, r)$ has to be true in each individual on the chain, it suffices to check that each approach to loop this chain breaks the disjointness. □

Even though equality and disjointness might appear to be duals, this belief is quickly crashed: equality is much harder and easily leads to undecidability.

Proposition 11. *Deciding satisfiability in $\mathcal{ALC}(\circ)^{\mathbf{e}}_{S}$ is undecidable and $\mathcal{ALC}(*,\circ)^{\mathbf{e}}_{S}$ does not have the finite model property.*

Proof. The undecidability result directly follows from results for Description Logics with role value maps [30]. An easy way to also see why the equality feature leads to undecidability is the following constraint set, which encodes a grid.

$$s \leftarrow \mathsf{eq}(ur, d) \wedge \mathsf{eq}(ru, d) \wedge \exists r.s \wedge \exists u.s \wedge \forall r.s \wedge \forall u.s$$

For the lack of finite model property, consider the following shapes graph $(\mathcal{C}, \mathcal{G})$:

$$\mathcal{C} = \{s \leftarrow \mathsf{eq}(r^*, t) \wedge \neg\mathsf{eq}(r^+, t) \wedge \exists r.s\},$$

and set $\mathcal{G} = \{s(a)\}$. Clearly, the infinite chain of r's, with t the reflexive and transitive closure of r, in which every individual is labelled with an s is an infinite model. In fact, it must be possible to homomorphically map this chain into any interpretation that validates $(\mathcal{C}, \mathcal{G})$. As $\mathsf{eq}(r^*, t) \wedge \neg\mathsf{eq}(r^+, t)$ has to be true in each individual on the chain, it suffices to check that each approach to loop this chain breaks successful validation. □

It looks much better when solely allowing '∪' in the equality and disjointness axioms: (finite) satisfiability in $\mathcal{ALC}(\cup)^{\mathbf{d},\mathbf{e}}_{S}$ is ExpTime-complete. In fact, this holds for much stronger fragments.

Proposition 12. *Deciding satisfiability in $\mathcal{ALCIQ}(\cup)^{\mathbf{d},\mathbf{e}}_{S}$, and (finite) satisfiability in $\mathcal{ALCOQ}(\cup)^{\mathbf{d},\mathbf{e}}_{S}$ and $\mathcal{ALCOI}(\cup)^{\mathbf{d},\mathbf{e}}_{S}$ is ExpTime-complete, and the latter two fragments have the finite model property.*

Proof. Note that for R a union of roles, $\mathsf{eq}(R, r)$ may be reduced to $\forall((R \setminus r) \cup (r \setminus R)).\bot$, where $E \setminus E' := E^{\mathcal{I}} \setminus E'^{\mathcal{I}}$, and $\mathsf{disj}(R, r)$ to $\forall(R \cap r).\bot$. Thus, in case only '∪' is allowed, the equality and disjointness features reduce to simple roles, which means the above fragments can be reduced to the description logics $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, resp. $\mathcal{ZOI}$. For all these logics, satisfiability is known to be decidable in ExpTime [31]. Furthermore, $\mathcal{ZOQ}$, and $\mathcal{ZOI}$ have the finite model property [32]. □

We note that the results described in this paper do not provide a complete picture of all known decidability results in the DL setting.

7. Conclusion and Outlook

We looked at the tight connection between Description Logics and SHACL. In this way, we derived many new complexity results for deciding (finite) satisfiability in SHACL. Specifically, for the general satisfiability problem the picture looks quite complete: as far as the author knows, only some small fragments remain unclear, like $\mathcal{ALC}(*,\cup)^{\mathbf{e}}_{S}$, or $\mathcal{ALC}(*)^{\mathbf{d},\mathbf{e}}_{S}$. However, when looking at finite satisfiability, the status is quite the opposite: a lot of work remains to be done. Specifically in the setting of SHACL, one of the standard tools for managing concrete data sets, the latter case is of uttermost importance.

Another direction for future work is to look at different semantics: in this paper, we considered (finite) satisfiability under the supported model semantics. However, there are more possibilities to consider: for instance the stable-model, or well-founded semantics. As far as the author knows there are no known complexity results regarding satisfiability or containment for any semantics other than the supported model semantics, leaving a major gap. Specifically, as researching complexity of satisfiability and containment problems is essential for determining which semantics are suitable in optimised SHACL-based solutions.

Acknowledgments

I would like to thank Magdalena Ortiz and Bartosz Bednarczyk for their support in this project, and the anonymous reviewers for their feedback.

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101034440.

Declaration on Generative AI

The author has not employed any Generative AI tools.

References

[1] J. Corman, J. L. Reutter, O. Savkovic, Semantics and validation of recursive SHACL, in: D. Vrandecic, K. Bontcheva, M. C. Suárez-Figueroa, V. Presutti, I. Celino, M. Sabou, L. Kaffee, E. Simperl (Eds.), The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part I, volume 11136 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 318–336. URL: https://doi.org/10.1007/978-3-030-00671-6_19. doi:10.1007/978-3-030-00671-6_19.

[2] M. Andresel, J. Corman, M. Ortiz, J. L. Reutter, O. Savkovic, M. Simkus, Stable model semantics for recursive SHACL, in: Y. Huang, I. King, T. Liu, M. van Steen (Eds.), WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020, ACM / IW3C2, 2020, pp. 1570–1580. URL: https://doi.org/10.1145/3366423.3380229. doi:10.1145/3366423.3380229.

[3] S. Ahmetaj, M. Ortiz, A. Oudshoorn, M. Simkus, Reconciling SHACL and ontologies: Semantics and validation via rewriting, in: K. Gal, A. Nowé, G. J. Nalepa, R. Fairstein, R. Radulescu (Eds.), ECAI 2023 - 26th European Conference on Artificial Intelligence, volume 372 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2023, pp. 27–35. URL: https://doi.org/10.3233/FAIA230250. doi:10.3233/FAIA230250.

[4] C. Okulmus, M. Simkus, SHACL validation under the well-founded semantics, in: P. Marquis, M. Ortiz, M. Pagnucco (Eds.), Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning, KR 2024, Hanoi, Vietnam. November 2-8, 2024, 2024. URL: https://doi.org/10.24963/kr.2024/52. doi:10.24963/KR.2024/52.

[5] P. G. Omran, K. Taylor, S. J. R. Méndez, A. Haller, Learning SHACL shapes from knowledge graphs, Semantic Web 14 (2023) 101–121. URL: https://doi.org/10.3233/SW-223063. doi:10.3233/SW-223063.

[6] D. Fernández-Álvarez, J. E. L. Gayo, D. Gayo-Avello, Automatic extraction of shapes using shexer, Knowl. Based Syst. 238 (2022) 107975. URL: https://doi.org/10.1016/j.knosys.2021.107975. doi:10.1016/J.KNOSYS.2021.107975.

[7] K. Rabbani, M. Lissandrini, K. Hose, Extraction of validating shapes from very large knowledge graphs, Proc. VLDB Endow. 16 (2023) 1023–1032. URL: https://www.vldb.org/pvldb/vol16/p1023-rabbani.pdf. doi:10.14778/3579075.3579078.

[8] P. Pareti, G. Konstantinidis, F. Mogavero, Satisfiability and containment of recursive SHACL, J. Web Semant. 74 (2022) 100721. URL: https://doi.org/10.1016/j.websem.2022.100721. doi:10.1016/J.WEBSEM.2022.100721.

[9] P. Pareti, G. Konstantinidis, F. Mogavero, T. J. Norman, SHACL satisfiability and containment, in: J. Z. Pan, V. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne, L. Kagal (Eds.), The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part I, volume 12506 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 474–493. URL: https://doi.org/10.1007/978-3-030-62419-4_27. doi:10.1007/978-3-030-62419-4_27.

[10] P. Pareti, SHACL2FOL: an FOL toolkit for SHACL decision problems, CoRR abs/2406.08018 (2024). URL: https://doi.org/10.48550/arXiv.2406.08018. doi:10.48550/ARXIV.2406.08018. arXiv:2406.08018.

[11] M. Leinberger, P. Seifer, T. Rienstra, R. Lämmel, S. Staab, Deciding SHACL shape containment through description logics reasoning, in: J. Z. Pan, V. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne, L. Kagal (Eds.), The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part I, volume 12506

of *Lecture Notes in Computer Science*, Springer, 2020, pp. 366–383. URL: https://doi.org/10.1007/978-3-030-62419-4_21. doi:10.1007/978-3-030-62419-4_21.
[12] B. Bogaerts, M. Jakubowski, J. V. den Bussche, SHACL: A description logic in disguise, in: G. Gottlob, D. Inclezan, M. Maratea (Eds.), Logic Programming and Nonmonotonic Reasoning - 16th International Conference, LPNMR 2022, Genova, Italy, September 5-9, 2022, Proceedings, volume 13416 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 75–88. URL: https://doi.org/10.1007/978-3-031-15707-3_7. doi:10.1007/978-3-031-15707-3_7.
[13] B. Bogaerts, M. Jakubowski, J. V. den Bussche, Expressiveness of SHACL features and extensions for full equality and disjointness tests, Log. Methods Comput. Sci. 20 (2024). URL: https://doi.org/10.46298/lmcs-20(1:16)2024. doi:10.46298/LMCS-20(1:16)2024.
[14] A. Oudshoorn, M. Ortiz, M. Simkus, Reasoning with the core chase: the case of SHACL validation over ELHI knowledge bases, in: L. Giordano, J. C. Jung, A. Ozaki (Eds.), Proceedings of the 37th International Workshop on Description Logics (DL 2024), Bergen, Norway, June 18-21, 2024, volume 3739 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024. URL: https://ceur-ws.org/Vol-3739/paper-7.pdf.
[15] W3C, Shape constraint language (SHACL), Technical Report. (2017). https://www.w3.org/TR/shacl.
[16] M. Krötzsch, OWL 2 profiles: An introduction to lightweight ontology languages, in: T. Eiter, T. Krennwallner (Eds.), Reasoning Web. Semantic Technologies for Advanced Query Answering - 8th International Summer School 2012, Vienna, Austria, September 3-8, 2012. Proceedings, volume 7487 of *Lecture Notes in Computer Science*, Springer, 2012, pp. 112–183. URL: https://doi.org/10.1007/978-3-642-33158-9_4. doi:10.1007/978-3-642-33158-9_4.
[17] O. Savkovic, E. Kharlamov, S. Lamparter, Validation of SHACL constraints over KGs with OWL 2 QL ontologies via rewriting, in: P. Hitzler, M. Fernández, K. Janowicz, A. Zaveri, A. J. G. Gray, V. López, A. Haller, K. Hammar (Eds.), The Semantic Web - 16th International Conference, ESWC 2019, Portorož, Slovenia, June 2-6, 2019, Proceedings, volume 11503 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 314–329. URL: https://doi.org/10.1007/978-3-030-21348-0_21. doi:10.1007/978-3-030-21348-0_21.
[18] S. Rudolph, Foundations of description logics, in: A. Polleres, C. d'Amato, M. Arenas, S. Handschuh, P. Kroner, S. Ossowski, P. F. Patel-Schneider (Eds.), Reasoning Web. Semantic Technologies for the Web of Data - 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures, volume 6848 of *Lecture Notes in Computer Science*, Springer, 2011, pp. 76–136. URL: https://doi.org/10.1007/978-3-642-23032-5_2. doi:10.1007/978-3-642-23032-5_2.
[19] C. Lutz, C. Areces, I. Horrocks, U. Sattler, Keys, nominals, and concrete domains, J. Artif. Intell. Res. 23 (2005) 667–726. URL: https://doi.org/10.1613/jair.1542. doi:10.1613/JAIR.1542.
[20] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, 2003.
[21] C. Lutz, U. Sattler, L. Tendera, The complexity of finite model reasoning in description logics, Inf. Comput. 199 (2005) 132–171. URL: https://doi.org/10.1016/j.ic.2004.11.002. doi:10.1016/J.IC.2004.11.002.
[22] S. Tobies, The complexity of reasoning with cardinality restrictions and nominals in expressive description logics, J. Artif. Intell. Res. 12 (2000) 199–217. URL: https://doi.org/10.1613/jair.705. doi:10.1613/JAIR.705.
[23] S. Tobies, Complexity results and practical algorithms for logics in knowledge representation, Ph.D. thesis, RWTH Aachen University, Germany, 2001. URL: http://sylvester.bth.rwth-aachen.de/dissertationen/2001/082/01_082.pdf.
[24] I. Pratt-Hartmann, Complexity of the two-variable fragment with counting quantifiers, J. Log. Lang. Inf. 14 (2005) 369–395. URL: https://doi.org/10.1007/s10849-005-5791-1. doi:10.1007/S10849-005-5791-1.
[25] R. Berger, The undecidability of the domino problem, 66, American Mathematical Soc., 1966.
[26] F. Baader, U. Sattler, Expressive number restrictions in description logics, J. Log. Comput. 9 (1999) 319–350. URL: https://doi.org/10.1093/logcom/9.3.319. doi:10.1093/LOGCOM/9.3.319.
[27] I. Horrocks, U. Sattler, S. Tobies, Practical reasoning for very expressive description logics, Log. J.

IGPL 8 (2000) 239–263. URL: https://doi.org/10.1093/jigpal/8.3.239. doi:10.1093/JIGPAL/8.3.239.

[28] F. Baader, U. Sattler, Number restrictions on complex roles in description logics: A preliminary report., in: KR, 1996, pp. 328–339.

[29] S. Göller, M. Lohrey, C. Lutz, PDL with intersection and converse: satisfiability and infinite-state model checking, J. Symb. Log. 74 (2009) 279–314. URL: https://doi.org/10.2178/jsl/1231082313. doi:10.2178/JSL/1231082313.

[30] M. Schmidt-Schauß, Subsumption in KL-ONE is undecidable, in: R. J. Brachman, H. J. Levesque, R. Reiter (Eds.), Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning (KR'89). Toronto, Canada, May 15-18 1989, Morgan Kaufmann, 1989, pp. 421–431.

[31] D. Calvanese, T. Eiter, M. Ortiz, Regular path queries in expressive description logics with nominals, in: C. Boutilier (Ed.), IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009, 2009, pp. 714–720. URL: http://ijcai.org/Proceedings/09/Papers/124.pdf.

[32] B. Bednarczyk, E. Kieronski, Finite entailment of local queries in the Z family of description logics, in: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022, AAAI Press, 2022, pp. 5487–5494. URL: https://doi.org/10.1609/aaai.v36i5.20487. doi:10.1609/AAAI.V36I5.20487.