

Nominal Schemas in Description Logics: Complexities Clarified

Markus Krötzsch and Sebastian Rudolph

Technische Universität Dresden, Germany

{markus.kroetzsch, sebastian.rudolph}@tu-dresden.de

Abstract

Nominal schemas extend description logics (DLs) with a restricted form of variables, thus integrating rule-like expressive power into standard DLs. They are also one of the most recently introduced DL features, and in spite of many works on algorithms and implementations, almost nothing is known about their computational complexity and expressivity. We close this gap by providing a comprehensive analysis of the reasoning complexities of a wide range of DLs—from \mathcal{EL} to \mathcal{SROIQ} —extended with nominal schemas. Both combined and data complexities increase by one exponential in most cases, with the one previously known case of \mathcal{SROIQ} being the main exception. Our proofs employ general modeling techniques that exploit the power of nominal schemas to succinctly represent many axioms, and which can also be applied to study DLs beyond those we consider. To further improve our understanding of nominal schemas, we also investigate their semantics, traditionally based on finite grounding, and show that it can be extended to infinite sets of individuals without affecting reasoning complexities. We argue that this might be a more suitable semantics when considering entailments of axioms with nominal schemas.

1 Introduction

The fruitful integration of reasoning on both schema and instance level poses a continued challenge to knowledge representation and reasoning. While description logics (DLs) excel at the former task, rule-based formalisms are often more adequate for the latter. An established and highly productive strand of research therefore continues to investigate ways of reconciling both paradigms.

A practical breakthrough in this area was the discovery of *DL-safe rules*, which ensure decidability of reasoning by restricting the applicability of rules to a finite set of elements that are denoted by an individual name (Motik, Sattler, and Studer 2005). As of today, DL-safe rules are the most widely used DL-rule extension, supported by several mainstream reasoners (Kołowski, Parsia, and Sirin 2006; Motik, Shearer, and Horrocks 2009).

More recently, *nominal schemas* have been proposed as an even tighter integration of “DL-safe” instance reasoning with DL schema reasoning (Krötzsch et al. 2011). A *nominal* is a DL concept expression $\{a\}$ that represents a singleton set containing only (the individual denoted by) a . Nominal schemas replace a by a variable x that ranges over all individual names, so that it might represent arbitrary nominals

$\{a\}$, where all occurrences of $\{x\}$ in one axiom represent the same nominal. For example,

$$\exists \text{hasFather}.\{x\} \sqcap \exists \text{hasMother}.\{\{y\} \sqcap \exists \text{married}.\{x\}\}$$

represents the set of all individuals whose father (x) and mother (y) are married to each other, where the parents must be represented by individual names. No standard DL can express this in such a concise way. The interplay with other DL features also makes nominal schemas more expressive than the combination of DLs and DL-safe rules (cf. Sect. 8).

Nominal schemas have thus caused significant research interest, and several reasoning algorithms that exploit this succinct representation have been proposed (Krisnathi and Hitzler 2012; Wang and Hitzler 2013; Steigmiller, Glimm, and Liebig 2013; Martínez, Wang, and Hitzler 2013). Most recently, Steigmiller, Glimm, and Liebig (2013) demonstrated that such algorithms can even outperform other systems for reasoning with DL-safe rules.

Surprisingly and in sharp contrast to these successes, many basic questions about the expressivity and complexity of nominal schemas have remained unanswered. A naive reasoning approach is based on replacing nominal schemas by nominals in all possible ways, which leads to complexity upper bounds one exponential above the underlying DL. The only tight complexity result so far is that the N2EXPTIME combined complexity of reasoning in the DL \mathcal{SROIQ} is not affected by nominal schemas—a result that reveals almost nothing about the computational or expressive impact of nominal schemas in general (Krötzsch et al. 2011). Beyond this singular result, it is only known that nominal schemas can simulate Datalog rules of any arity using \exists , \sqcap , and the universal role U (Knorr, Hitzler, and Maier 2012).

In this paper, we give a comprehensive account of the reasoning complexities of a wide range of DLs, considering both combined complexities (w.r.t. the size of the given knowledge base) and data complexities (w.r.t. the size of the ABox only). Figure 1 summarizes our results for combined complexities for DLs with nominal schemas (right; marked by the letter \mathcal{V}) in comparison with known complexities of DLs with nominals (left). It turns out that \mathcal{SROIQ} is an exception, while most other DLs experience exponential complexity increases due to nominal schemas.

The effects on the data complexity are even more striking. The data complexity of standard DLs is either in P (for \mathcal{EL} and Horn-DLs, which restrict the use of \sqcup and \neg) or in NP . In contrast, the data complexities for all nominal-schema DLs in Fig. 1 are only one exponential below their combined complexity, i.e., EXPTIME or NEXPTIME for most cases.

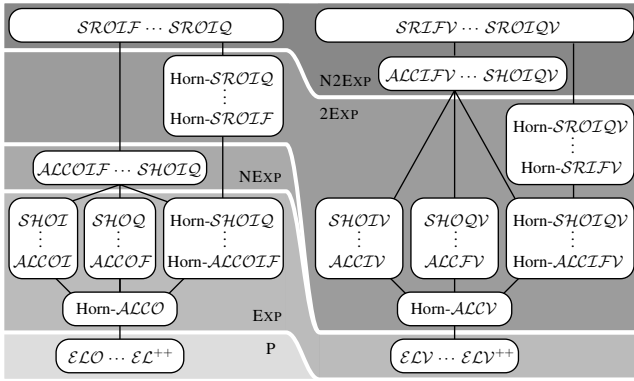


Figure 1: Combined complexities for DLs with nominals compared to DLs with nominal schemas

To obtain these results, we identify general modeling techniques that use nominal schemas to express complex schema information very succinctly. Two fundamental techniques provide the basis for most of our hardness proofs:

- **TBox-to-ABox Internalization** A TBox is replaced by a small set of “template axioms” with nominal schemas, and the original TBox is expressed with ABox assertions.
- **GCI Iterators** Templates of TBox axioms (general concept inclusions, short GCIs) are instantiated by replacing placeholder concepts by concepts from an exponentially long list of “indexed” concept names.

TBox-to-ABox internalization explains why the data complexity of most DLs with nominal schemas agrees with the combined complexity of their underlying standard DL. $SROIQV$ is a noteworthy exception where the internalization is not possible. GCI iterators are a kind of generalized TBox axiom that can be used to encode exponentially large TBoxes polynomially using nominal schemas. This technique can be applied to TBoxes from known hardness proofs to boost complexities by one exponential. Both techniques provide concrete illustrations for the expressive power of nominal schemas and outline ways to obtain results for DLs that we did not consider.

After establishing these results, we revisit the formal semantics of nominal schemas. Normally, nominal schemas are considered to represent a finite set of nominals, based on individuals that either occur in the knowledge base or are part of some finite signature. This can lead to unintuitive effects, since entailments may become invalid when adding more individuals. We thus study the semantics obtained when using an infinite set of individual names instead. This makes it impossible to replace nominal schemas by nominals in all possible ways to decide entailment. Surprisingly, reasoning is still decidable with the same complexity results. Indeed, the consequences of both approaches turn out to agree under some mild assumptions.

2 Standard Description Logics

The DLs we consider are based on the well-known DL $SROIQ$ (Horrocks, Kutz, and Sattler 2006). The precise

syntax and semantics that we adhere to is explained in detail in the freely available DL Primer (Krötzsch, Simančík, and Horrocks 2012), so we will not repeat it here. We also consider Horn-DLs, which, in essence, forbid \sqcup and \neg , and restrict the use of \forall (Krötzsch, Rudolph, and Hitzler 2013).

Our results apply to many DLs. The smallest DL we consider is \mathcal{EL} , allowing only \sqcap , \exists , and \top . We do not consider variants of DL-Lite, since these DLs typically exclude nominals, and are therefore no natural candidates for extension with nominal schemas. We also exclude lightweight approaches like DLP, pD^* , and OWL RL, since the rule-like expressiveness of these DLs is subsumed completely by nominal schemas (see Section 4). We thus use the term *standard description logic* to mean a DL \mathcal{L} with $\mathcal{EL} \subseteq \mathcal{L} \subseteq SROIQ$. Our complexity results refer to standard reasoning tasks, and this is what the term *reasoning* will refer to. Formally, a *standard reasoning problem* of \mathcal{L} is any problem that can be polynomially reduced to checking if an \mathcal{L} knowledge base KB entails a fact $A(a)$ (note that \mathcal{EL} consistency checking is trivial). Nondeterministic complexity classes may occur in two versions, e.g., NEXPTIME and co-NEXPTIME, depending on the exact reasoning task.

Two knowledge bases are semantically equivalent if they have the same models. This is often too restrictive, since many “essentially equivalent” transformations make use of auxiliary symbols that affect the set of models. A useful way to describe that a knowledge base faithfully captures the semantics of another one is to use *conservative extensions*.

Definition 1. Consider a knowledge base KB over a signature Σ , and a knowledge base KB' over a signature Σ' that extends Σ . Then KB' is a (model-)conservative extension of KB if (a) every model of KB' is a model of KB, and (b) every model of KB can be extended to a model of KB' by adding suitable interpretations for additional signature symbols. Note that we do not require $KB \subseteq KB'$.

Essentially all mainstream DLs allow axioms to be normalized. We exploit this in some of our proofs.

Definition 2. A DL \mathcal{L} is called *normalizable*, if there is a finite set $\{\alpha_1, \dots, \alpha_k\}$ of \mathcal{L} axioms and a polynomially computable function τ that maps \mathcal{L} knowledge bases to \mathcal{L} knowledge bases such that for every \mathcal{L} knowledge base KB

- $\tau(KB)$ is a conservative extension of KB, and
- every axiom of $\tau(KB)$ can be obtained from some $\alpha \in \{\alpha_1, \dots, \alpha_k\}$ by renaming signature symbols.

3 Nominal Schemas: Syntax and Semantics

Recall that a $SROIQ$ knowledge base is based on a signature consisting of finite sets N_I of individual names, N_C of concept names, and N_R of role names. To define nominal schemas, we extend signatures by an additional finite set N_V of variable names. A nominal schema is a concept of the form $\{x\}$ with $x \in N_V$, which can be used in axioms and complex concept expressions like any other $SROIQ$ concept. DLs with nominal schemas are denoted by their standard name extended by \mathcal{V} as in $SROIQV$ or \mathcal{ELV} .

The semantics of nominal schemas are defined by *grounding*. For a $SROIQV$ axiom α , $\text{ground}(\alpha)$ is the set of

all $SROIQ$ axioms obtained from α by uniformly replacing all variables in α by individual names in N_I in all possible ways. The grounding of a knowledge base KB is $\text{ground}(\text{KB}) := \bigcup_{\alpha \in \text{KB}} \text{ground}(\alpha)$. Note that $\text{ground}(\text{KB})$ is finite, since N_I is finite. An interpretation satisfies a $SROIQV$ knowledge base KB if it satisfies $\text{ground}(\text{KB})$, and satisfiability and entailment are defined accordingly.

The semantics of nominal schemas depends on the set N_I of individual names in the given signature. For example,

$$\{a\} \sqsubseteq \{b\} \models \{a\} \sqsubseteq \{x\} \quad (1)$$

is a valid entailment in a signature with $N_I = \{a, b\}$, but not in one with $N_I = \{a, b, c\}$. However, the choice of N_I (as long as it is non-empty) is only relevant when asking if axioms with nominal schemas are entailed but not for standard reasoning problems, such as instance or subsumption checking. Hence we often do not mention N_I explicitly, though, in general, always assume that the signature is given. Moreover, instances of entailment problems are always considered in the context of a single signature that covers all involved axioms.

One could take other approaches, which are worth mentioning. First, instead of using N_I , one could define nominal schemas to refer to all individual names that occur in the knowledge base. This raises questions when considering entailment problems (Which of the knowledge bases? Both?), and it leads to a kind of non-monotonicity, where $\{a\} \sqsubseteq \{x\}$ follows from $\{a\} \sqsubseteq \{b\}$, while the logically weaker axiom $\{a\} \sqsubseteq \{x\} \sqcup \{c\}$ does not (since it extends the range of $\{x\}$). Secondly, to overcome this problem, one could restrict the range of each nominal schema explicitly by requiring additional syntax. The original formulation of DL-safe rules, e.g., requires each rule variable x to appear in an atom of the form $O(x)$ in the premise, and O was defined in the ABox (Motik, Sattler, and Studer 2005). For DL axioms, however, such restrictions would unduly clutter the syntax. Finally, instead of considering N_I to be variable and finite, one could consider an infinite signature used by all knowledge bases. Grounding is no longer possible in this approach, and it is not the semantics that was assumed in previous works. We study this approach in Section 8, and find that it leads to mostly the same results as the traditional formulation.

Regarding reasoning complexity, it is important that we do not assume N_I to be fixed. Rather, the signature is part of the problem instance of an entailment problem. Therefore, the *size of a problem* always includes the size of the signature. Analogously, *data complexity* refers to the complexity of a problem under the assumption that the size of TBox and RBox is bounded by a constant, while the ABox and the signature can vary. When speaking of the size of a knowledge base or of an ABox, we always assume that the size of the signature is included.

Since the size of $\text{ground}(\text{KB})$ is exponential in the size of KB and polynomial in the size of the ABox (where the number of variables per TBox axiom is bounded), we obtain some general upper bounds for the complexity of reasoning.

Theorem 1. *Consider a DL \mathcal{L} . The combined complexity of \mathcal{LV} is at most exponentially higher than that of \mathcal{LO} . If the combined complexity of \mathcal{LO} is in a complexity class \mathcal{C} that subsumes \mathcal{P} , then the data complexity of \mathcal{LV} is in \mathcal{C} as well.*

4 Expressiveness of Nominal Schemas

We now take a closer look at the expressive power of nominal schemas, and draw some first conclusions about the complexity of reasoning. It was immediately observed that nominal schemas can be used to encode DL-safe rules (Krötzsch et al. 2011), and this result has later been extended to show that they can express extended DL-safe rules with predicates of any arity (Knorr, Hitzler, and Maier 2012). The universal role U was used in both encodings. However, it is easy to see that we can remove this requirement. To represent a first-order logic atom $p(t_1, \dots, t_n)$, where $t_i \in N_I \cup N_V$, we use a fresh concept A_p and auxiliary roles $\text{atom}, \text{arg}_1, \dots, \text{arg}_n$. We define $\text{enc}(p(t_1, \dots, t_n))$ to be the formula

$$\exists \text{atom}.(A_p \sqcap \exists \text{arg}_1.\{t_1\} \sqcap \dots \sqcap \exists \text{arg}_n.\{t_n\}). \quad (2)$$

A rule $B_1 \wedge \dots \wedge B_\ell \rightarrow H$, where B_1, \dots, B_ℓ, H are first-order atoms, can now be encoded by the DL axiom

$$\text{enc}(B_1) \sqcap \dots \sqcap \text{enc}(B_\ell) \sqsubseteq \text{enc}(H) \quad (3)$$

(as usual, the left-hand side is \top if $\ell = 0$). It is easy to see that this captures the semantics of an arbitrary set of function-free Horn rules (a.k.a. Datalog). Hence, the simple DL \mathcal{ELV} can already capture Datalog and thus inherits its EXPTIME complexity (Dantsin et al. 2001). The matching upper bound follows from Theorem 1 and the polynomial combined complexity of \mathcal{EL} (Baader, Brandt, and Lutz 2005).

Theorem 2. *Standard reasoning in \mathcal{ELV} is EXPTIME-complete for combined complexity.*

Note that the encoding of rules does not provide any interaction between unary and binary rule predicates and DL concepts and roles, respectively. We could add this using axioms like $\exists \text{atom}.(A_r \sqcap \exists \text{arg}_1.\{x\} \sqcap \exists \text{arg}_2.\{y\}) \sqsubseteq \exists \text{aux}.(\{x\} \sqcap \exists r.\{y\})$. However, some DL knowledge bases may interfere with the correctness of the encoding if they restrict the overall domain size. We address this issue in the next section.

In addition to the above, it is intuitively clear that nominal schemas are at least as expressive as nominals. The next definition provides a suitable construction. As before, we only need features of \mathcal{EL} .

Definition 3. *Let KB be a knowledge base. The knowledge base $\text{nomelim}(\text{KB})$ is obtained from KB as follows:*

- *replace all occurrences of $\{a\}$ in KB by O_a , where O_a is a fresh concept name,*
- *add an ABox axiom $O_a(a)$ for all $a \in N_I$ in KB ,*
- *add the axiom $\top \sqsubseteq \exists \text{somenom}.\{x\}$, where somenom is a fresh role name, and*
- *add the axiom $O_a \sqcap \exists \text{somenom}.\{x\} \sqcap O_a \sqsubseteq \{x\}$ for each of the fresh O_a .*

Essentially this procedure installs atomic concepts O_a which are axiomatized to have the same extension as the nominals they substitute. It is clear that one can compute $\text{nomelim}(\text{KB})$ in polynomial time. Moreover, $\text{nomelim}(\text{KB})$ is a conservative extension of KB . Part (a) of Definition 1 is straightforward since $\{a\} \equiv O_a$ is a direct consequence of $\text{nomelim}(\text{KB})$ for every nominal $\{a\}$ of KB . For (b) we enrich any model \mathcal{I} of KB by $O_a^{\mathcal{I}} = \{a^{\mathcal{I}}\}$ and $\text{somenom}^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \{a^{\mathcal{I}} \mid a \text{ individual in } \text{KB}\}$ to find a model of $\text{nomelim}(\text{KB})$. This leads to the following general result.

Theorem 3. *For any DL \mathcal{L} with $\mathcal{EL} \subseteq \mathcal{L}$, the combined and data complexities of standard reasoning in \mathcal{LO} are lower bounds for the respective complexities of \mathcal{LV} .*

Nominal schemas thus generalize the expressiveness of nominals, DL-safe rules, and (DL-safe) Datalog. Nevertheless, using grounding, all of these can be expressed with nominals already, though an exponential loss in efficiency is unavoidable in general. DL-safe rules, in contrast, cannot even express nominals. The Datalog encoding (3) illustrates a type of expression that is not expressible in DLs extended with DL-safe rules. Indeed, if we consider all rules and axioms as first-order logic formulae, then the existential quantifiers in the conclusion of (3) occur in the scope of (i.e., “depend on”) an arbitrary number of nominal schema assignments. In contrast, existential quantifiers in DLs (with or without DL-safe rules) occur only in axioms that share exactly one variable between the left and right formula. In other words, using Skolemization to eliminate quantifiers, one only needs unary Skolem functions in standard DLs, but Skolem functions of arbitrary arity with nominal schemas.

A related property of DLs with nominal schemas is that they cannot be normalized in the sense of Definition 2. Our complexity results allow us to show the following theorem at the end of Section 7; the case of *SRIOQ* remains open.

Theorem 4. *No DL \mathcal{LV} with $\mathcal{ELV} \subseteq \mathcal{LV} \subseteq \mathcal{SHOIQV}$ is normalizable.*

5 Domain Size Matters

Several of our results are based on transforming a knowledge base by using fresh individuals to represent classes. For example, for a concept *Car*, we can introduce an individual c_{Car} and express a fact *Car*(a) as $\text{type}(a, c_{\text{Car}})$. However, DLs that support nominals can interfere with this encoding.

Example 1. *Consider a knowledge base KB containing the TBox axiom $\top \sqsubseteq \{a, b\}$ and the ABox axioms $A(a), \neg B(a), \neg A(b), B(b), C(a), C(b)$. Clearly, KB is satisfiable. If we represent classes by individuals as above, we obtain the knowledge base KB' with ABox axioms $\text{type}(a, c_A), \neg \text{type}(a, c_B), \neg \text{type}(b, c_A), \text{type}(b, c_B), \text{type}(a, c_C), \text{type}(b, c_C)$. Since every model \mathcal{I} of KB' has to map any of c_A, c_B, c_C to either $a^{\mathcal{I}}$ or $b^{\mathcal{I}}$, KB' is unsatisfiable.*

To address this obstacle, we show how an \mathcal{L} reasoning problem can be transformed in such a way that it is legitimate to focus attention on models with unbounded domains. The following notion will come handy for this.

Definition 4. *We call a knowledge base KB unbounded if, for every model \mathcal{I} of KB, there is a model \mathcal{J} of KB with infinite domain such that $\mathcal{I} \models \alpha$ iff $\mathcal{J} \models \alpha$ for every atomic concept assertion α .*

Unboundedness is guaranteed in some situations. For example, every *SRIOQ* knowledge base not using the universal role is unbounded (Mehdi and Rudolph 2011, Lemma 1), and every Horn-*SRIOQ* knowledge base that has a model with a domain size > 1 is unbounded as well. In the remainder of this section, we show how to transform bounded knowledge bases into unbounded ones. We can then focus on unbounded knowledge bases in the rest of the paper.

Concept expressions of the form $\forall U.C$ complicate the conversion of a knowledge base into an unbounded one, especially if the availability of other modeling features cannot be assumed. However, one can remove such expressions in a way that (next to using the universal role U which is present anyway) only requires features of \mathcal{EL} .

Definition 5. *Let $\text{remuniv}(\text{KB})$ be the knowledge base obtained from KB by performing the following steps for each concept expression of the form $\forall U.C$ contained in KB, where $A_{\forall U.C}$ and $Aux_{\forall U.C}$ are fresh concept names:*

- substitute all occurrences of $\forall U.C$ by $A_{\forall U.C}$
- add the following axioms to KB:

$$\exists U.A_{\forall U.C} \sqsubseteq C \quad (4)$$

$$\top \sqsubseteq \exists U.Aux_{\forall U.C} \quad (5)$$

$$\exists U.(Aux_{\forall U.C} \sqcap C) \sqsubseteq A_{\forall U.C} \quad (6)$$

Lemma 5. *$\text{remuniv}(\text{KB})$ is a conservative extension of KB.*

Proof. First, we show that every model of $\text{remuniv}(\text{KB})$ is a model of KB by observing that $\text{remuniv}(\text{KB}) \models A_{\forall U.C} \equiv \forall U.C$ for all the replaced expressions $\forall U.C$. To this end, we make a detour via first-order logic. We start with the Axiom (4) and obtain $\exists U.A_{\forall U.C} \sqsubseteq C \Rightarrow \forall x.(\exists y.(A_{\forall U.C}(y) \rightarrow C(x)) \Rightarrow (\exists y.(A_{\forall U.C}(y) \rightarrow \forall x.C(x)) \Rightarrow \forall y.(A_{\forall U.C}(y) \rightarrow \forall x.C(x)) \Rightarrow A_{\forall U.C} \sqsubseteq \forall U.C$.

For the other direction, we start from Axiom (6), $\exists U.(Aux_{\forall U.C} \sqcap C) \sqsubseteq A_{\forall U.C}$, translate it into first-order logic: $\forall x.(\exists y.(Aux_{\forall U.C}(y) \wedge C(y)) \rightarrow A_{\forall U.C}(x))$, transform it into $\forall x.(\forall y.(\neg Aux_{\forall U.C}(y) \vee \neg C(y)) \vee A_{\forall U.C}(x))$, and further into $\forall y.(\neg Aux_{\forall U.C}(y) \vee \neg C(y) \vee \forall x.(A_{\forall U.C}(x)))$ to finally arrive at $\forall y.(Aux_{\forall U.C}(y) \rightarrow (\neg C(y) \vee \forall x.(A_{\forall U.C}(x))))$. Since the FOL version of Axiom (5) is simply $\exists y.Aux_{\forall U.C}(y)$, we can use this to instantiate the premise of the last rule and thus obtain: $\exists y.(\neg C(y) \vee \forall x.(A_{\forall U.C}(x))) \Rightarrow \forall x.(\exists y.(\neg C(y)) \vee A_{\forall U.C}(x)) \Rightarrow \forall x.(\neg \forall y.(C(y)) \vee A_{\forall U.C}(x)) \Rightarrow \forall x.(\forall y.(C(y)) \rightarrow A_{\forall U.C}(x)) \Rightarrow \forall U.C \sqsubseteq A_{\forall U.C}$.

Next, we have to show that every model \mathcal{I} of KB can be extended to a model \mathcal{J} of $\text{remuniv}(\text{KB})$ by providing appropriate extensions for the fresh concept names. This can be achieved via $A_{\forall U.C}^{\mathcal{J}} := (\forall U.C)^{\mathcal{I}}$ and by letting $Aux_{\forall U.C}^{\mathcal{J}} := \top^{\mathcal{I}}$ if $(\neg C)^{\mathcal{I}} = \emptyset$ and $Aux_{\forall U.C}^{\mathcal{J}} := (\neg C)^{\mathcal{I}}$ otherwise. \square

Since remuniv is clearly polynomial, we do not have to consider expressions $\forall U.C$ below. Our next transformation to produce an unbounded knowledge base is reminiscent of a technique used to allow the introduction of new individuals for metamodeling (Glimm, Rudolph, and Völker 2010).

Definition 6. *Let KB be an \mathcal{L} knowledge base for a standard DL \mathcal{L} , without expressions of the form $\forall U.C$. We define $\text{unbound}(\text{KB})$ as follows, where Top is a fresh concept:*

1. Recursively, outside-in, we replace all concept subexpressions C in all axioms by $\text{Top} \sqcap C$.
2. If \mathcal{L} allows for universal quantification, we add a GCI $\top \sqsubseteq \forall r.\text{Top}$ for every (possibly inverse) role $r \neq U$ in KB.
3. For every individual name a in KB, we add $\text{Top}(a)$; if no individual occurs, we introduce a fresh one a_{new} .

Example 2. Consider our knowledge base KB from Example 1; $\text{unbound}(\text{KB})$ contains the axioms $\text{Top} \sqsubseteq \text{Top} \sqcap \{a, b\}$, $\text{Top}(a)$, $\text{Top}(b)$, $\text{Top} \sqcap A(a)$, $\text{Top} \sqcap \neg(\text{Top} \sqcap B)(a)$, $\text{Top} \sqcap \neg(\text{Top} \sqcap A)(b)$, $\text{Top} \sqcap B(b)$, $\text{Top} \sqcap C(a)$, $\text{Top} \sqcap C(b)$.

Obviously, the described transformation can be computed in polynomial time and outputs a knowledge base of polynomial size. Moreover, for any standard DL \mathcal{L} , and any \mathcal{L} knowledge base, $\text{unbound}(\text{KB})$ is an \mathcal{L} knowledge base.

Lemma 6 (Properties of unbound).

1. For any knowledge base KB and any model \mathcal{J} of $\text{unbound}(\text{KB})$, there is a model \mathcal{I} of KB such that $\mathcal{I} \models A(a)$ if and only if $\mathcal{J} \models \text{Top} \sqcap A(a)$ for every atomic concept assertion $A(a)$.
2. For any model \mathcal{I} of a knowledge base KB, there is a model \mathcal{J} of $\text{unbound}(\text{KB})$ with an infinite domain such that $\mathcal{I} \models A(a)$ if and only if $\mathcal{J} \models \text{Top} \sqcap A(a)$ for every atomic concept assertion $A(a)$.
3. $\text{unbound}(\text{KB})$ is unbounded.
4. $\text{KB} \models A(a)$ if and only if $\text{unbound}(\text{KB}) \models \text{Top} \sqcap A(a)$.

Proof. Claim (1). We provide a construction of \mathcal{I} . Given a model $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ of $\text{unbound}(\text{KB})$ we define $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ as follows: $\Delta^{\mathcal{I}} := \text{Top}^{\mathcal{J}}$ (note that nonemptiness is guaranteed since $\text{Top}^{\mathcal{J}}$ contains at least one named individual by definition of unbound), $a^{\mathcal{I}} := a^{\mathcal{J}}$ for all $a \in N_I$, $A^{\mathcal{I}} := A^{\mathcal{J}} \cap \Delta^{\mathcal{I}}$ for $A \in N_C$, and $r^{\mathcal{I}} := r^{\mathcal{J}} \cap (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}})$ for $r \in N_R$. We can now show by induction on the structure of the concept expressions that $C^{\mathcal{I}} = \text{topbound}(C)^{\mathcal{J}}$ for any concept expression C occurring in KB, where $\text{topbound}(C)$ denotes the result of replacing, recursively and outside-in, all concept subexpressions D of C (including C itself) by $\text{Top} \sqcap D$. Base cases and induction steps are straightforward for all cases except for universal quantification.

Thus, consider $C = \forall r.D$, with $\text{topbound}(C) = \text{Top} \sqcap \forall r.\text{topbound}(D)$ and assume $D^{\mathcal{I}} = \text{topbound}(D)^{\mathcal{J}}$ holds by induction hypothesis. Since C occurs in KB, the latter must be formulated in a DL supporting universal quantification, thus $\text{unbound}(\text{KB})$ must by definition contain the axiom $\top \sqsubseteq \forall r.\text{Top}$. Hence $y \in \text{Top}^{\mathcal{J}}$ for all $\langle x, y \rangle \in r^{\mathcal{J}}$. Now we observe equivalence of the following statements: $x \in C^{\mathcal{I}} \iff x \in (\forall r.D)^{\mathcal{I}} \iff x \in \Delta^{\mathcal{I}}$ and $\forall y$ with $\langle x, y \rangle \in r^{\mathcal{I}}$ holds $y \in D^{\mathcal{I}} \iff x \in \text{Top}^{\mathcal{J}}$ and $\forall y$ with $\langle x, y \rangle \in r^{\mathcal{J}} \cap (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}})$ holds $y \in \text{topbound}(D)^{\mathcal{J}} \iff x \in \text{Top}^{\mathcal{J}}$ and $\forall y$ with $\langle x, y \rangle \in r^{\mathcal{J}} \cap (\text{Top}^{\mathcal{J}} \times \text{Top}^{\mathcal{J}})$ holds $y \in \text{topbound}(D)^{\mathcal{J}} \iff x \in \text{Top}^{\mathcal{J}}$ and $\forall y$ with $\langle x, y \rangle \in r^{\mathcal{J}}$ holds $y \in \text{topbound}(D)^{\mathcal{J}} \iff x \in (\text{Top} \sqcap \forall r.\text{topbound}(D))^{\mathcal{J}} \iff x \in \text{topbound}(C)^{\mathcal{J}}$.

Having established that $C^{\mathcal{I}} = \text{topbound}(C)^{\mathcal{J}}$, we can show that satisfaction of all axioms from $\text{unbound}(\text{KB})$ in \mathcal{J} directly ensures satisfaction of all axioms from KB in \mathcal{I} . We also find, for any individual name a and concept name A , that: $\mathcal{I} \models A(a) \iff a^{\mathcal{I}} \in A^{\mathcal{I}} \iff a^{\mathcal{I}} \in A^{\mathcal{J}} \cap \Delta^{\mathcal{I}} \iff a^{\mathcal{J}} \in A^{\mathcal{J}} \cap \text{Top}^{\mathcal{J}} \iff \mathcal{J} \models \text{Top} \sqcap A(a)$ as claimed.

Claim (2). We assume a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of KB and construct $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ as follows: $\Delta^{\mathcal{J}} := \Delta^{\mathcal{I}} \cup N_I$; $a^{\mathcal{J}} := a^{\mathcal{I}}$ for all $a \in N_I$, where $a_{\text{new}}^{\mathcal{J}} \in \Delta^{\mathcal{I}}$ if a_{new} has been introduced in $\text{unbound}(\text{KB})$; $A^{\mathcal{J}} := A^{\mathcal{I}}$ for all $A \in N_C$ interpreted by \mathcal{I} ; $\text{Top}^{\mathcal{J}} := \Delta^{\mathcal{I}}$; and $r^{\mathcal{J}} := r^{\mathcal{I}}$ for $r \in N_R$. Again, we can use induction to show that the correspondence $C^{\mathcal{I}} =$

$\text{topbound}(C)^{\mathcal{J}}$ holds and conclude that all axioms from $\text{unbound}(\text{KB})$ are valid in \mathcal{J} . Clearly, \mathcal{J} has an infinite domain. We also obtain for any concept name A and individual name a the following equivalence: $\mathcal{J} \models \text{Top} \sqcap A(a) \iff a^{\mathcal{J}} \in A^{\mathcal{J}} \cap \text{Top}^{\mathcal{J}} \iff a^{\mathcal{I}} \in A^{\mathcal{I}} \cap \Delta^{\mathcal{I}} \iff \mathcal{I} \models A(a)$.

Claim (3). This is a direct consequence of the preceding two. For an arbitrary model \mathcal{J} of $\text{unbound}(\text{KB})$ we create the model \mathcal{I} of KB as in the proof of Claim (1). From \mathcal{I} , we then construct a model \mathcal{J}' of $\text{unbound}(\text{KB})$ as in the proof of Claim (2). Then we know that \mathcal{J} and \mathcal{J}' satisfy the same atomic concept assertions and \mathcal{J}' has an infinite domain.

Claim (4). This is equivalent to the statement that $\text{KB} \not\models A(a)$ iff $\text{unbound}(\text{KB}) \not\models \text{Top} \sqcap A(a)$. This can be easily shown by seeing that any model of KB not satisfying $A(a)$ can be turned into a model of $\text{unbound}(\text{KB})$ not satisfying $\text{Top} \sqcap A(a)$ (by Claim (2)) and vice versa (by Claim (1)). \square

Example 3. Referring to Example 2, we find that $\text{unbound}(\text{KB})$ is satisfiable and, unlike KB, it even has a model \mathcal{J} with infinite domain $\Delta^{\mathcal{J}} = \{a^{\mathcal{J}}, b^{\mathcal{J}}\} \cup N_I$. Moreover we find, e.g., the correspondence between KB entailing $A \sqsubseteq C$ and $\text{unbound}(\text{KB})$ entailing $\text{Top} \sqcap A \sqsubseteq \text{Top} \sqcap C$.

6 Data Complexities

From Theorem 1 and the known complexity results for various DLs, we obtain the following upper bounds for the data complexity: P for \mathcal{ELV}^{++} (Baader, Brandt, and Lutz 2005), EXPTIME for \mathcal{SHOIV} , \mathcal{SHOQV} , and Horn- \mathcal{SHOIQV} (Hladik 2004; Glimm, Horrocks, and Sattler 2008; Ortiz, Rudolph, and Simkus 2010), NEXPTIME for \mathcal{SHOIQV} (Tobies 2001), 2EXPTIME for Horn- \mathcal{SROIQV} (Ortiz, Rudolph, and Simkus 2010), and N2EXPTIME for \mathcal{SROIQV} (Kazakov 2008).¹ We can obtain matching hardness results in many cases, even for much simpler DLs.

Theorem 7. 1. The data complexity of \mathcal{ELV} is P-hard.

2. The data complexity of Horn- \mathcal{ALCV} is EXPTIME-hard.

3. The data complexity of \mathcal{ALCIFV} is (co)NEXPTIME-hard.

For \mathcal{ELV} , this already follows from the hardness of data complexity for \mathcal{EL} . To prove the other cases, we apply a generic method of expressing TBoxes as ABoxes.

Definition 7. Consider a DL \mathcal{L} with $\mathcal{EL} \subseteq \mathcal{L} \subseteq \mathcal{SROIQ}$ and an \mathcal{L} TBox axiom $\alpha = C \sqsubseteq D$. The template for α , denoted $\text{tmpl}(\alpha)$, is defined as follows. Let $\sigma_1, \dots, \sigma_n$ be a list of all individual names and concept names in α . Let A_α be a fresh concept name, and let gci , type , and symb_i ($1 \leq i \leq n$) be fresh role names. Then $\text{tmpl}(\alpha)$ is the \mathcal{LV} axiom

$$\exists \text{gci}. (A_\alpha \sqcap \exists \text{symb}_1. \{x_1\} \sqcap \dots \sqcap \text{symb}_n. \{x_n\}) \sqcap C' \sqsubseteq D'$$

where C' and D' are obtained from C and D , respectively, by replacing each concept name σ_i by $\exists \text{type}. \{x_i\}$ and each individual name σ_j by x_j .

¹Most complexity upper bounds for DLs above \mathcal{ALCOIN} are only known to hold when assuming unary coding of numbers; our results inherit this restriction.

The template instance for α , denoted $\text{tins}(\alpha)$, is the following set of ABox assertions:

$$\{A_\alpha(c_\alpha), \text{symb}_1(c_\alpha, c_{\sigma_1}), \dots, \text{symb}_n(c_\alpha, c_{\sigma_n})\}$$

where c_α and $c_{\sigma_1}, \dots, c_{\sigma_n}$ are fresh individual names.

We extend temp and tins to sets of axioms by taking the union of the results, and to ABox axioms $A(c)$ and $r(c, d)$ by expressing them as TBox axioms $\{c\} \sqsubseteq A$ and $\{c\} \sqsubseteq \exists r.\{d\}$, respectively. For an unbounded knowledge base KB with ABox \mathcal{A} , TBox \mathcal{T} , and RBox \mathcal{R} , we define $\text{tt}(\text{KB})$ as the set $\{\top \sqsubseteq \exists \text{gci}.\{x\}\} \cup \text{temp}(\mathcal{A} \cup \mathcal{T}) \cup \text{tins}(\mathcal{A} \cup \mathcal{T}) \cup \mathcal{R}$, where we assume that the same individuals c_σ for representing a symbol σ in any translation.

It is easy to verify the following correctness property, where the axiom $\top \sqsubseteq \exists \text{gci}.\{x\}$ ensures that the template instantiation individuals c_α are universally reachable via gci .

Lemma 8. $\text{KB} \models A(c)$ iff $\text{tt}(\text{KB}) \models \exists \text{type}.\{c_A\}(c)$.

This yields a polynomial reduction from one reasoning problem to another. We can improve this to obtain a reduction where the size of the TBox of $\text{tt}(\text{KB})$ is not affected by the size of N_I, N_C , or the TBox of KB. Indeed, since $\text{tmpl}(\alpha)$ contains only role names and variable names, we only need one template for each variant of α obtained by uniformly renaming concepts and individuals. In particular, if the given DL admits a finite number of normal forms, then it is enough to include one template for each axioms that can be obtained by uniform renaming of roles in some normal form. If N_R is fixed, this yields a fixed TBox and also a fixed RBox (since a fixed N_R only allows for a bounded number of RBox axioms in normal form). Summing up, all standard reasoning problems for a normalizable DL \mathcal{L} that can be expressed with a fixed, finite set of role names N_R can be expressed in \mathcal{LV} using a fixed, finite TBox and RBox.

Lemma 9. Let $\mathcal{L}[N_R]$ be the DL \mathcal{L} restricted to signatures that use a fixed, finite set N_R of role names. If $\mathcal{L}[N_R]$ is normalizable using only roles from N_R , and its combined complexity is hard for a complexity class C that includes P , then so is the data complexity of \mathcal{LV} .

This allows us to show Theorem 7. Indeed, the known hardness proofs for Horn- \mathcal{ALC} (Krötzsch, Rudolph, and Hitzler 2013) and \mathcal{ALCOIF} (Tobies 2000) are based on constructing a sequence of knowledge bases that use a fixed, finite set of role names. Hence, the respective complexities are the claimed lower bounds for the data complexities of Horn- \mathcal{ALCV} and \mathcal{ALCIFV} .

Interestingly, Lemma 9 is not applicable to \mathcal{SROIQV} , since the known N2EXPTIME-hardness proof for \mathcal{SROIQ} relies on an unbounded set of role names (Kazakov 2008). Indeed, it turns out that our N2EXPTIME upper bound for the data complexity of \mathcal{SROIQV} can be improved to NEXPTIME, which is tight according to Theorem 7. With a very similar argument one can establish an EXPTIME upper bound for data complexity of Horn- \mathcal{SROIQV} .

Theorem 10. The data complexity of \mathcal{SROIQV} is in NEXPTIME, assuming unary coding of numbers. The data complexity of Horn- \mathcal{SROIQV} is in EXPTIME.

Proof. In order to show our claim, we deploy a two-step reduction and analyze its effects on the knowledge base. Starting from the \mathcal{SROIQV} knowledge base $\text{KB} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$, we obtain the \mathcal{SROIQ} knowledge base $\text{KB}' = (\mathcal{A}, \mathcal{T}', \mathcal{R})$ by grounding KB. Next, we obtain the $\mathcal{ALCHOIQ}$ knowledge base $\text{KB}'' = (\mathcal{A}, \mathcal{T}'', \emptyset)$ by applying the RBox removal transformation introduced by Kazakov (2008). Analyzing the grounding step, we find that $\text{size}(\mathcal{T}') \leq \text{size}(\mathcal{T}) + \text{size}(\mathcal{A})^{\text{size}(\mathcal{T})}$. Moreover, for the RBox removal step, we obtain $\text{size}(\mathcal{T}'') \leq 2^{\text{size}(\mathcal{R})} \cdot \text{size}(\mathcal{T}')$. Therefore we get $\text{size}(\text{KB}'') = \text{size}(\mathcal{A}) + \text{size}(\mathcal{T}'') \leq \text{size}(\mathcal{A}) + 2^{\text{size}(\mathcal{R})} \cdot \text{size}(\mathcal{T}') \leq \text{size}(\mathcal{A}) + 2^{\text{size}(\mathcal{R})} \cdot (\text{size}(\mathcal{T}) + \text{size}(\mathcal{A})^{\text{size}(\mathcal{T})})$. Since \mathcal{T} and \mathcal{R} are fixed, we find that KB'' is polynomial in the size of \mathcal{A} . Since the combined complexity of standard reasoning in $\mathcal{ALCHOIQ}$ is in NEXPTIME for unary coding of numbers (Tobies 2001), we have proven our claim. The same size argument as above holds for the transformation of Horn- \mathcal{SROIQV} into Horn- \mathcal{SROIQ} (via grounding) and further into Horn- $\mathcal{ALCHOIFb}$, which has a combined complexity of EXPTIME (Ortiz, Rudolph, and Simkus 2010). This shows the claim for Horn- \mathcal{SROIQ} . \square

7 Combined Complexities

We already know the combined complexities of \mathcal{ELV} (EXPTIME, cf. Section 4) and \mathcal{SROIQV} (N2EXPTIME). The N2EXPTIME argument for \mathcal{SROIQV} (Krötzsch et al. 2011) can be directly applied to Horn- \mathcal{SROIQV} (Ortiz, Rudolph, and Simkus 2010) to establish 2EXPTIME combined complexity. From Theorem 1 and the known complexity results we further obtain that the upper bound for the combined complexity of \mathcal{SHOIQV} , and \mathcal{SHOQV} is 2EXPTIME (Hladik 2004; Glimm, Horrocks, and Sattler 2008). The upper bounds Theorem 1 yields for \mathcal{SHOIQV} and Horn- \mathcal{SHOIQV} coincide with those of \mathcal{SROIQV} and Horn- \mathcal{SROIQV} , respectively. It turns out that these bounds are tight: reasoning in DLs above Horn- \mathcal{ALCV} is 2EXPTIME-hard, and reasoning in DLs above \mathcal{ALCIFV} is (co)N2EXPTIME-hard.

To show these results, we employ a general technique of encoding “axiom templates” that can encode an exponential number of concept inclusion axioms in polynomial space. We call this expressive feature *GCI iterator* since it iterates over a list of indexed concept names to produce concrete axioms. For example, the GCI iterator $A[\mathbf{i}] \sqsubseteq A[\mathbf{i} + 1]$ [$\mathbf{i} = 1, \dots, n$] encodes n axioms of the form $A_i \sqsubseteq A_{i+1}$. We will show that GCI iterators can be expressed polynomially using nominal schemas, even when assuming binary coding of the number n . This allows us to encode exponentially large knowledge bases using polynomially many axioms with nominal schemas. In particular, we will do this for the specific knowledge bases that have been used to show the known hardness results for Horn- \mathcal{ALC} and \mathcal{ALCOIF} , thus boosting the complexity by one exponential.

Definition 8. Consider a DL signature $\langle N_I, N_C, N_R \rangle$. A GCI iterator over this signature is an expression

$$C \sqsubseteq D [\mathbf{i} = 1, \dots, n]$$

where $n \geq 1$ and $C \sqsubseteq D$ is a general concept inclusion over

$\langle N_I, N_C \cup \{A[1], \dots, A[n+1], A[i], A[i+1] \mid A \in N_C\}, N_R \rangle$.
 Note that i is a literal part of the syntax, not a placeholder for a specific number. The additional concept names $A[\dots]$ are assumed to be distinct from all concepts in N_C .

The expansion of a GCI iterator is the set of GCIs over $\langle N_I, N_C \cup \{A[1], \dots, A[n+1] \mid A \in N_C\}, N_R \rangle$ obtained by replacing, for each $i \in \{1, \dots, n\}$, all concepts $A[i]$ by $A[i]$, and all concepts $A[i+1]$ by $A[i+1]$.

For a DL \mathcal{L} , we let \mathcal{L}_{GI} be \mathcal{L} extended by GCI iterators as axioms. The semantics of an \mathcal{L}_{GI} knowledge base KB is given by the translation into \mathcal{L} through replacing all GCI iterators by their expansions, denoted $\text{expand}(\text{KB})$.

Note that, assuming binary encoding of the value n , GCI iterators allow for an exponentially more succinct representation of their expansion.

Next, we will show that standard reasoning in a DL \mathcal{L}_{GI} without nominal schemas can be polynomially reduced to standard reasoning in \mathcal{LV} . We restrict attention to the case where the upper bound n in all GCI iterators is of the form 2^ℓ for some $\ell \geq 1$. Arbitrary n could be supported with some additional effort, but this is not relevant for our proofs.

Definition 9. Consider $\ell \geq 1$, role names $\text{bit}_1, \dots, \text{bit}_\ell$, and individual names 0 and 1. For a list b_1, \dots, b_ℓ with $b_i \in \{0, 1\} \cup N_V$, we define $\text{Bits}(b_1, \dots, b_\ell) := \prod_{i=1}^\ell \exists \text{bit}_i. \{b_i\}$. Consider additional role names num , this , and next . KB_{num} consists of the following axioms, for all $1 \leq i \leq \ell$:

$$\begin{aligned} \top &\sqsubseteq \exists \text{num}. \exists \text{this}. \text{Bits}(0, \dots, 0) \\ \exists \text{this}. \text{Bits}(x_1, \dots, x_i, 0, 1, \dots, 1) &\equiv \exists \text{next}. \text{Bits}(x_1, \dots, x_i, 1, 0, \dots, 0) \\ \exists \text{num}. \exists \text{next}. \text{Bits}(x_1, \dots, x_\ell) &\equiv \exists \text{num}. \exists \text{this}. \text{Bits}(x_1, \dots, x_\ell) \end{aligned}$$

Given a GCI iterator γ of the form $C \sqsubseteq D \ [i = 1, \dots, 2^\ell]$, we define $\text{vexpand}(\gamma)$ to be the axiom $N \sqcap C' \sqsubseteq D'$, where $N = \exists \text{num}. (\exists \text{this}. \text{Bits}(x_1, \dots, x_\ell) \sqcap \exists \text{next}. \text{Bits}(y_1, \dots, y_\ell))$ and C' and D' are obtained from C and D , respectively, by the following replacements:

- $A[i]$ is replaced by $\exists \text{type}. (A' \sqcap \text{Bits}(x_1, \dots, x_\ell))$,
- $A[i+1]$ is replaced by $\exists \text{type}. (A' \sqcap \text{Bits}(y_1, \dots, y_\ell))$,
- $A[k]$ is replaced by $\exists \text{type}. (A' \sqcap \text{Bits}(b_1, \dots, b_\ell))$ where b_1, \dots, b_ℓ is the binary encoding of k ,

where A' is a fresh concept name for each $A \in N_C$.

For an unbounded \mathcal{L}_{GI} knowledge base $\text{KB} = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ where all GCI iterators range from 1 to 2^ℓ , the \mathcal{LV} knowledge base $\text{itov}(\text{KB})$ is obtained by adding the axioms KB_{num} and replacing each GCI iterator $\gamma \in \text{KB}$ with $\text{vexpand}(\gamma)$.

Theorem 11. For any DL \mathcal{L} , itov provides a polytime reduction from standard reasoning in \mathcal{L}_{GI} to standard reasoning in \mathcal{LV} : for every unbounded \mathcal{L}_{GI} knowledge base KB and DL axiom $A(c)$, we have $\text{KB} \models A(c)$ iff $\text{itov}(\text{KB}) \models \text{itov}(A(c))$.

Proof. We write $\text{bin}(k)$ for the list of ℓ bits of a number $k < 2^\ell$ in binary encoding. The minimal model for KB_{num} is one where every individual has one num -successor in concept $\exists \text{this}. \text{Bits}(\text{bin}(k))$ for each number $0 \leq k < 2^\ell$. Moreover, if $k < 2^\ell - 1$ (i.e., $\text{bin}(k)$ contains a 0), the corresponding successor also is in the class $\exists \text{next}. \text{Bits}(\text{bin}(k+1))$.

Now every model $\mathcal{I} \models \text{expand}(\text{KB})$ with $\mathcal{I} \not\models A(c)$ can be transformed into a model $\mathcal{I}' \models \text{itov}(\text{KB})$ with $\mathcal{I}' \not\models$

$\text{itov}(A(c))$. By unboundedness, we can assume that the domain of \mathcal{I} contains infinitely many individuals. We can thus find distinct individuals to represent bits and numbers, and we interpret num , this , next , and bit_i according to the minimal model. Moreover, for all $A \in N_C$ and $0 \leq k < 2^\ell$, we add an element $\delta_{A[k]}$ with $\delta_A \in (A' \sqcap \text{Bits}(\text{bin}(k)))^{\mathcal{I}'}$, and we set $\langle \delta, \delta_{A[k]} \rangle \in \text{type}$ iff $\delta \in A[k]^{\mathcal{I}'}$. Roles and concepts that are not of the form $B[k]$ are interpreted as in \mathcal{I} . It is easy to see that \mathcal{I}' has the claimed properties.

For the converse, note that the minimal model of KB_{num} is indeed found in every model of $\text{itov}(\text{KB})$. Thus, if $\mathcal{I}' \models \text{itov}(\text{KB})$ with $\mathcal{I}' \not\models \text{itov}(A(c))$, then there is more general interpretation $\mathcal{I}'' \models \text{itov}(\text{KB})$ with $\mathcal{I}'' \not\models \text{itov}(A(c))$ where KB_{num} is interpreted as in the minimal model. Using similar correspondences as before, we can construct a suitable model $\mathcal{I} \models \text{expand}(\text{KB})$ with $\mathcal{I} \not\models A(c)$. \square

We now use Theorem 11 to show additional hardness results by using GCI iterators to re-encode known hardness proofs for exponentially large knowledge bases.

Theorem 12. The combined complexity of Horn- \mathcal{ALCV} is 2EXPTIME-hard.

Proof. The known EXPTIME-hardness result for Horn- \mathcal{ALC} is based on reducing the problem of deciding if a word w is accepted by an alternating Turing machine (ATM) in polynomial space to the problem if a Horn- \mathcal{ALC} knowledge base entails a GCI, where both knowledge base and GCI are of polynomial size w.r.t. $|w|$ (Krötzsch, Rudolph, and Hitzler 2013). One can generalize this construction to arbitrary (possibly non-polynomial) space bounds s , leading to knowledge base of size polynomial in s . We modify the encoding to be able to express this knowledge base in Horn- \mathcal{ALC}_{GI} , so that the resulting knowledge base is logarithmic in the size of s . Thus, we can simulate exponential space ATM runs with a polynomial Horn- \mathcal{ALC}_{GI} knowledge base.

Let s be an arbitrary space bound. To encode an s -space ATM $\langle Q, \Sigma, \Delta, q_0 \rangle$, the original encoding uses the following concept names: A_q for $q \in Q$ (ATM is in state q), H_i for $i = 1, \dots, s$ (ATM head is at position i), $C_{\sigma, i}$ with $\sigma \in \Sigma$ and $i = 1, \dots, s$ (tape position i contains symbol σ), and A (ATM accepts this configuration). Accordingly, we use concepts A_q , $H[i]$, $C_\sigma[i]$, and A . Roles S_δ encode δ -successor configurations. The original encoding of ATM transitions $\delta = \langle q, \sigma, q', \sigma', r \rangle$ can then be expressed as follows:

$$A_q \sqcap H[i] \sqcap C_\sigma[i] \sqsubseteq \exists S_\delta. (A_{q'} \sqcap H[i+1] \sqcap C_{\sigma'}[i]) \ [i=1, \dots, s]$$

Transitions with leftwards head movement are expressed analogously. To apply Theorem 11, we use the same range for all GCI iterators. This has the side effect that the tape has cells $1, \dots, s+1$, while i ranges over $1, \dots, s$ and $i+1$ ranges over $2, \dots, s+1$; thus, when using i in axioms that refer to all tape positions, we generally assume that there is also one more axiom for covering $s+1$ as well.

The accepting conditions are modelled by $A_q \sqcap \exists S_\delta. A \sqsubseteq A$ (for existential states q) and $A_q \sqcap H[i] \sqcap C_\sigma[i] \sqcap \prod_{\delta \in \bar{\Delta}} (\exists S_\delta. A) \sqsubseteq A \ [i=1, \dots, s]$ (for universal states q with

possible transitions $\tilde{\Delta}$ for symbol σ). To ensure preservation of the ATM memory between configurations, the original proof uses axioms $H_j \sqcap C_{\sigma,i} \sqsubseteq \forall S_{\delta}. C_{\sigma,i}$ for all $i \neq j$. This cannot be expressed with GCI iterators. Instead, we introduce new concepts $L[i]$ (cell i is left of the head) and $R[i]$ (cell i is right of the head), which we can axiomatize using GCI iterators $H[i] \sqsubseteq R[i+1]$, $R[i] \sqsubseteq R[i+1]$, $H[i+1] \sqsubseteq L[i]$, and $L[i+1] \sqsubseteq L[i]$, all with the same range as before. The rules of memory preservation can now be expressed with the GCI iterators $L[i] \sqcap C_{\sigma,i} \sqsubseteq \forall S_{\delta}. C_{\sigma,i}$ and $R[i] \sqcap C_{\sigma,i} \sqsubseteq \forall S_{\delta}. C_{\sigma,i}$, for all $\delta \in \Delta$ and $\sigma \in \Sigma$.

Finally, the initialization of the tape in the original proof simply defines the tape contents for each position. To initialize the $s - |w|$ many blank cells after the input word, we use concepts $B[i]$ and axioms $B[i] \sqsubseteq B[i+1]$ and $B[i] \sqsubseteq C_{\square}[i]$. The initial configuration is now described with the concept

$$I_w := A_{q_0} \sqcap H_1 \sqcap C_{\sigma_1}[1] \sqcap \dots \sqcap C_{\sigma_{|w|}}[|w|] \sqcap B[|w|+1].$$

Let $\text{KB}_{\mathcal{M},w}$ be the knowledge base that contains all of the above GCIs. It is a direct consequence from (Krötzsch, Rudolph, and Hitzler 2013) that $\text{KB}_{\mathcal{M},w} \models I_w \sqsubseteq A$ if the ATM \mathcal{M} accepts w in space s . Since GCI iterators can encode the range $[i=1, \dots, s]$ in space logarithmic in s , reasoning with a polynomially large $\text{KB}_{\mathcal{M},w}$ can solve the word problem of exponential space ATMs, which is 2EXPTIME -hard. The claim thus follows from Theorem 11. \square

Theorem 13. *The combined complexity of $\mathcal{ALCCIFV}$ is (co)N2EXPTIME-hard.*

Proof. We proceed with the case of \mathcal{ALCCIF} . We build on the proof of Tobies (2000), which reduced the tiling problem for exponential grids to \mathcal{ALCCIF} consistency checking. To construct an exponential grid, one encodes coordinates of tiles in binary numbers, represented by concept names X_1, \dots, X_{ℓ} (horizontal position) and Y_1, \dots, Y_{ℓ} (vertical position), respectively. We will express these by indexed concepts $X[i]$ and $Y[i]$. Horizontal neighbours are connected by a role h , vertical ones by a role v . Functionality restrictions and inverse functionality restrictions on h and v ensure uniqueness of neighbours. The concept with the maximal coordinates (right upper corner) is subsumed by a nominal, ensuring that the potentially diverging structure of h and v successors collapses into one grid.

We need to express that the encoded position of the right (and upper) neighbour is the successor. For numbers of linear length, this can be done as in Definition 9; to allow for numbers of exponential length, we use auxiliary concepts $\hat{X}[i]$ to express that all bits up to position $i-1$ are set to true, i.e., that the conjunction $\prod_{j=1}^{i-1} X[j]$ holds. $\hat{Y}[i]$ plays the same role for vertical positions. The propagation of X -coordinates can now be expressed with the following GCI iterators, all of which we assume to have range $[i = 1, \dots, \ell]$ for some ℓ :

$$\top \sqsubseteq \exists h. \top \sqcap \leq 1 h. \top \sqcap \leq 1 h^{-}. \top \quad (7)$$

$$\top \sqsubseteq \hat{X}[1] \quad \hat{X}[i] \sqcap X[i] \equiv \hat{X}[i+1] \quad (8)$$

$$\hat{X}[i] \sqsubseteq (X[i] \sqcap \forall h. \neg X[i]) \sqcup (\neg X[i] \sqcap \forall h. X[i]) \quad (9)$$

$$\neg \hat{X}[i] \sqsubseteq (X[i] \sqcap \forall h. X[i]) \sqcup (\neg X[i] \sqcap \forall h. \neg X[i]) \quad (10)$$

$$X[i] \sqsubseteq \forall v. X[i] \quad \neg X[i] \sqsubseteq \forall v. \neg X[i] \quad (11)$$

Axiom (7) ensures a chain of h successors. Axioms (8) express the relation of $\hat{X}[i]$ and $X[i]$ explained above. Axioms (9) and (10) characterize the binary increment of numbers. Axioms (11) ensure that horizontal positions are preserved among vertical neighbours. We assume the same formulation for the vertical propagation, swapping h with v and X with Y , respectively. Note that $i+1$ only occurs in (8); indeed, the bits $X[\ell+1]$ and $Y[\ell+1]$, which exist due to our uniform choice of range, have no significance in our modelling. To “start” the grid, we assert that there is a lower left corner: $\{a\} \sqsubseteq \neg X[i] \sqcap \neg Y[i]$. To “close” the grid, we add a nominal to the upper right corner: $\hat{X}[\ell+1] \sqcap \hat{Y}[\ell+1] \sqsubseteq \{o\}$. Note that the grid will continue to have successors beyond this point, due to axiom (7), but these do not need to form a grid and will not affect whether the tiling problem has a solution (one can always ensure that there are tiles to continue in non-grid structures only).

Formally, a tiling problem is given by a set of tiles $\mathcal{D} = \{D_1, \dots, D_n\}$ together with horizontal and vertical compatibility relations $V, H \subseteq \mathcal{D} \times \mathcal{D}$. Using concept names D_i (which are not iterated over), we can axiomatize correct tilings by the axioms $\top \sqsubseteq \bigsqcup_{1 \leq i \leq d} (D_i \sqcap \prod_{j \neq i} \neg D_j)$, $D_i \sqsubseteq \forall v. \bigsqcup_{(D_i, D_j) \in V} D_j$, and $D_i \sqsubseteq \forall h. \bigsqcup_{(D_i, D_j) \in H} D_j$. It is easy to see that this construction reduces the tiling problem on a grid of size $2^{\ell} \times 2^{\ell}$ to checking the consistency of an $\mathcal{ALCCIF}_{\text{GI}}$ knowledge base of size polynomial in $\log(\ell)$ (since ℓ is encoded in binary in GCI iterators). Since the double-exponential tiling problem is N2EXPTIME-hard, the claim follows from Theorem 11. \square

Proof of Theorem 4. As the number of nominal schemas in normalized axioms is bounded, a normalized knowledge base can be grounded polynomially. Thus, the combined complexity of any normalizable DL \mathcal{LV} coincides with the combined complexity of \mathcal{L} . It is known that $\text{P} \subseteq \text{EXPTIME} \subseteq 2\text{EXPTIME}$ and that $\text{NEXPTIME} \subseteq \text{N2EXPTIME}$. Together with our previous results, this shows the claim. \square

8 An Alternative Semantics

As discussed in Section 3, our semantics of nominal schemas depends on the given set N_I of individual names. If we allow N_I to be empty, then this can even affect the entailment of standard DL axioms. For example, $\top \sqsubseteq \exists r. \{x\} \not\models \top \sqsubseteq \exists r. \top$ if $N_I = \emptyset$. However, in most cases, only entailments of axioms with nominal schemas are affected, as discussed for (1) in Section 3. Such a dependence on the set of “known” symbols might be considered to be in conflict with the open-world assumption. The ontology language OWL considers a potentially infinite signature of URIs (W3C OWL Working Group 27 October 2009).²

To address this problem, one could modify the semantics of nominal schemas by assuming the set N_I to be infinite and fixed. A severe practical consequence of this suggestion is that $\text{ground}(\alpha)$ and $\text{ground}(\text{KB})$ become infinite if they contain any nominal schemas. While this does not pose problems for any of the definitions as such, we seem to lose

²On the other hand, OWL also provides ways to explicitly declare a finite set of signature symbols in an ontology.

the upper complexity bounds of Theorem 1. However, we will show that the alternative semantics does not actually affect reasoning complexities, and in fact agrees with the standard semantics in many cases.

In the rest of this section, we assume N_I to be infinite, and use \bar{N} for a set $\bar{N} \subseteq N_I$ that corresponds to our earlier finite signature. We use $\models_{\bar{N}}$ to denote standard entailment w.r.t. \bar{N} , and \models_{∞} to denote entailment w.r.t. the infinite N_I .

Theorem 14. *For any standard DL \mathcal{L} , any \mathcal{LV} knowledge base KB, any \mathcal{LV} axiom and any finite set \bar{N} which contains more individual names than the sum of the number of individual names used in $\text{KB} \cup \alpha$ and the number of variable names in α , we have $\text{KB} \models_{\bar{N}} \alpha$ iff $\text{KB} \models_{\infty} \alpha$.*

Proof. “If.” Assume $\text{KB} \models_{\infty} \alpha$. We pick a function $\psi : N_I \rightarrow \bar{N}$ such that $\psi(a) = a$ for all $a \in \bar{N}$. Note that such a ψ always exists since \bar{N} is nonempty by assumption. Lifting ψ to concept expressions, axioms and sets of axioms in the usual way, we observe that $\psi(\text{ground}_{\infty}(\text{KB})) = \text{ground}_{\bar{N}}(\text{KB})$, as well as $\psi(\text{ground}_{\infty}(\alpha)) = \text{ground}_{\bar{N}}(\alpha)$. Moreover, from $\text{ground}_{\infty}(\text{KB}) \models \beta$ follows that $\psi(\text{ground}_{\infty}(\text{KB})) \models \psi(\beta)$ for all $\beta \in \psi(\text{ground}_{\infty}(\alpha))$. Hence we obtain $\text{ground}_{\bar{N}}(\text{KB}) \models \text{ground}_{\bar{N}}(\alpha)$ and therefore $\text{KB} \models_{\bar{N}} \alpha$.

“Only if.” Assume $\text{KB} \models_{\bar{N}} \alpha$. Let $V = \{x_1, \dots, x_n\}$ be the set of nominal variables occurring in α . Then, by assumption, we find n distinct individuals $\{a_1, \dots, a_n\} \subseteq \bar{N}$ not occurring in $\text{KB} \cup \alpha$. Let $\alpha' = \alpha[x_1/a_1, \dots, x_n/a_n]$. Clearly $\text{KB} \models_{\bar{N}} \alpha'$. Then also $\text{KB} \models_{\infty} \alpha'$, since the grounding of KB with individuals from \bar{N} is contained in the grounding of KB with all individuals. In order to show $\text{ground}_{\infty}(\text{KB}) \models \alpha$, we will show that $\text{ground}_{\infty}(\text{KB}) \models \beta$ for all $\beta \in \text{ground}_{\infty}(\alpha)$. So, given such a β and the variable assignment val of $V = \{x_1, \dots, x_n\}$ to N_I that produces β from α , we pick a surjective mapping $\varphi : N_I \rightarrow N_I$ such that $\varphi(a) = a$ for every individual name mentioned in $\text{KB} \cup \alpha$ and $\varphi(a_i) = val(x_i)$ (this is possible since N_I is infinite). Lifting φ to concept expressions, axioms and sets of axioms in the usual way, it is obvious that $\text{KB} \models_{\infty} \alpha' \implies \text{ground}_{\infty}(\text{KB}) \models \alpha' \implies \varphi(\text{ground}_{\infty}(\text{KB})) \models \varphi(\alpha') \implies \text{ground}_{\infty}(\text{KB}) \models \varphi(\alpha') \implies \text{ground}_{\infty}(\text{KB}) \models \beta \implies \text{KB} \models_{\infty} \beta$. Note that $\varphi(\text{ground}_{\infty}(\text{KB})) = \text{ground}_{\infty}(\text{KB})$ by construction. \square

The previous proof requires that the unique name assumption (UNA) is not made, as is standard for all DLs considered in this paper. Moreover, imposing the unique name assumption in our setting would enforce all models to have an infinite domain, which makes domain-restricting axioms such as $\top \sqsubseteq \{a, b, c\}$ unsatisfiable.

Theorem 14 asserts that the two semantics coincide whenever the finite individual set is “large enough” for the considered knowledge base and tested consequence. For the special case of entailment of axioms without nominal schemas, it suffices if there is at least one individual name.

Corollary 15. *For any standard DL \mathcal{L} , \mathcal{LV} -reasoning complexities (both data and combined) for \models and \models_{∞} coincide.*

We have shown that entailment checking w.r.t. \models_{∞} can be reduced to entailment checking w.r.t. $\models_{\bar{N}}$ for a suitable \bar{N} . On the other hand, any such reasoning task w.r.t. $\models_{\bar{N}}$ can

be reduced to reasoning w.r.t. \models_{∞} if disjunction is available, since $\text{KB} \models_{\bar{N}} \alpha$ iff $\text{KB} \cup \{\{x\} \sqsubseteq \bigsqcup_{a \in \bar{N}} \{a\}\} \models_{\infty} \alpha$.

The alternative semantics also comes handy when analyzing expressivity, since one cannot finitely express nominal schemas using grounding. We can use this to show that \mathcal{LV} is generally more expressive than \mathcal{L} extended with DL-safe rules. Consider the axiom

$$\{x\} \sqsubseteq \exists r. \exists r. \exists r. \{x\}. \quad (12)$$

Now, suppose there exists a *SRIOQ* knowledge base KB and a set RB of DL-safe rules (where w.l.o.g. all used concept expressions are concept names) such that $\text{KB} \cup \text{RB}$ is a conservative extension of the above axiom. Now we pick an individual name a not mentioned in $\text{KB} \cup \text{RB}$ and define two interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ as follows: $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}} = \{b_1, b_2, b_3 \mid b \in N_I\} \cup \bar{N}$, $b^{\mathcal{I}} = b^{\mathcal{J}} = b_1$ for all $b \in N_I$, $r^{\mathcal{I}} = \{(b_1, b_2), (b_2, b_3), (b_3, b_1) \mid b \in N_I\}$, and $r^{\mathcal{J}} = r^{\mathcal{I}} \setminus \{(a_1, a_2), (a_2, a_3), (a_3, a_1)\}$. Note that \mathcal{I} is a model of the above axiom while \mathcal{J} is not. Consequently, by assumption there must be an extension \mathcal{I}' of \mathcal{I} satisfying $\text{KB} \cup \text{RB} \models \mathcal{I}'$ but any extension \mathcal{J}' of \mathcal{J} must not be a model of $\text{KB} \cup \text{RB}$. Now we pick a bijective function $\varphi : N_I \setminus \{a\} \rightarrow N_I$ with $\varphi(b) = b$ for all b mentioned in $\text{KB} \cup \text{RB}$ and use it to define a renaming ρ of $\Delta^{\mathcal{I}'}$ as follows: $\rho(b_i) := c_i$ with $c = \varphi(b)$ for all $b \in N_I \setminus \{a\}$ and $i \in \{1, 2, 3\}$, $\rho(a_i) := i$ for all $i \in \{1, 2, 3\}$ and $\rho(i) := i + 3$ for all $i \in \bar{N}$. Now let \mathcal{K} be the interpretation obtained from \mathcal{I}' via the renaming ρ . It is easy to check that \mathcal{K} is an extension of \mathcal{J} . We also find that \mathcal{K} must be a model of KB since \mathcal{K} and \mathcal{I}' are isomorphic as far as the used signature is concerned. Moreover, \mathcal{K} is a model of RB since assuming, for a rule $R \in \text{RB}$, a violating variable assignment var , $\varphi \circ var$ would be a violating assignment for \mathcal{I}' . Thus we have found an extension of \mathcal{J} that is a model of $\text{KB} \cup \text{RB}$, which contradicts our initial assumption. Therefore, the above axiom cannot be expressed by *SRIOQ* extended by DL-safe rules.

9 Conclusion and Future Work

We have clarified the data and combined complexities for nominal-schema-extensions of all DLs between \mathcal{EL} and *SRIOQ*. The techniques developed for this purpose (TBox-to-ABox internalization and GCI iterators) are generic and directly applicable to DLs outside this range. For example, it is an obvious consequence of our findings and the results from (Rudolph, Krötzsch, and Hitzler 2008b) that adding arbitrary Boolean role constructors on simple roles to any DL above *ALCIFV* does not increase data or combined complexity; the same holds if conjunctions on simple roles are added to \mathcal{ELV}^{++} . Adding the concept product \times (Rudolph, Krötzsch, and Hitzler 2008a) does not impact complexities either. Likewise, it is trivial to establish EXPTIME completeness (data) and 2EXPTIME completeness (combined) for *ZOQV* and *ZOV* (Calvanese, Eiter, and Ortiz 2009).

The most obvious avenue for future work is the investigation of the complexities for conjunctive query answering.

Acknowledgement This work was supported by the DFG in project DIAMOND (Emmy Noether grant KR 4381/1-1).

References

- [Baader, Brandt, and Lutz 2005] Baader, F.; Brandt, S.; and Lutz, C. 2005. Pushing the \mathcal{EL} envelope. In Kaelbling, L., and Safiotti, A., eds., *Proc. 19th Int. Joint Conf. on Artificial Intelligence (IJCAI'05)*, 364–369. Professional Book Center.
- [Brewka and Lang 2008] Brewka, G., and Lang, J., eds. 2008. *Proc. 11th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'08)*. AAAI Press.
- [Calvanese, Eiter, and Ortiz 2009] Calvanese, D.; Eiter, T.; and Ortiz, M. 2009. Regular path queries in expressive description logics with nominals. In Boutilier, C., ed., *Proc. 21st Int. Joint Conf. on Artificial Intelligence (IJCAI'09)*, 714–720. IJCAI.
- [Dantsin et al. 2001] Dantsin, E.; Eiter, T.; Gottlob, G.; and Voronkov, A. 2001. Complexity and expressive power of logic programming. *ACM Computing Surveys* 33(3):374–425.
- [Glimm, Horrocks, and Sattler 2008] Glimm, B.; Horrocks, I.; and Sattler, U. 2008. Unions of conjunctive queries in \mathcal{SHOQ} . In Brewka and Lang (2008), 252–262.
- [2010] Glimm, B.; Rudolph, S.; and Völker, J. 2010. Integrated metamodeling and diagnosis in OWL 2. In Patel-Schneider, P. F.; Pan, Y.; Glimm, B.; Hitzler, P.; Mika, P.; Pan, J.; and Horrocks, I., eds., *Proc. 9th Int. Semantic Web Conf. (ISWC'10)*, volume 6496 of *LNCS*, 257–272. Springer.
- [2004] Hladik, J. 2004. A tableau system for the description logic \mathcal{SHIO} . In Sattler, U., ed., *Contributions to the Doctoral Programme of the 2nd Int. Joint Conf. on Automated Reasoning (IJCAR'04)*, volume 106 of *CEUR WS Proceedings*. CEUR-WS.org.
- [2006] Horrocks, I.; Kutz, O.; and Sattler, U. 2006. The even more irresistible \mathcal{SROIQ} . In Doherty, P.; Mylopoulos, J.; and Welty, C. A., eds., *Proc. 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'06)*, 57–67. AAAI Press.
- [2008] Kazakov, Y. 2008. \mathcal{RIQ} and \mathcal{SROIQ} are harder than \mathcal{SHOIQ} . In Brewka and Lang (2008), 274–284.
- [2012] Knorr, M.; Hitzler, P.; and Maier, F. 2012. Reconciling OWL and non-monotonic rules for the Semantic Web. In Raedt, L. D.; Bessière, C.; Dubois, D.; Doherty, P.; Frasconi, P.; Heintz, F.; and Lucas, P. J. F., eds., *Proc. 20th European Conf. on Artificial Intelligence (ECAI'12)*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, 474–479. IOS Press.
- [2006] Kolovski, V.; Parsia, B.; and Sirin, E. 2006. Extending the $\mathcal{SHOIQ}(d)$ tableaux with DL-safe rules: First results. In Parsia, B.; Sattler, U.; and Toman, D., eds., *Proc. 19th Int. Workshop on Description Logics (DL'06)*, volume 198 of *CEUR WS Proceedings*. CEUR-WS.org.
- [2012] Krisnadhi, A., and Hitzler, P. 2012. A tableau algorithm for description logics with nominal schema. In Krötzsch, M., and Straccia, U., eds., *Proc. 6th Int. Conf. on Web Reasoning and Rule Systems (RR 2012)*, volume 7497 of *LNCS*, 234–237. Springer.
- [2011] Krötzsch, M.; Maier, F.; Krisnadhi, A. A.; and Hitzler, P. 2011. A better uncle for OWL: Nominal schemas for integrating rules and ontologies. In *Proc. 20th Int. Conf. on World Wide Web (WWW'11)*, 645–654. ACM.
- [2013] Krötzsch, M.; Rudolph, S.; and Hitzler, P. 2013. Complexities of Horn description logics. *ACM Trans. Comput. Logic* 14(1):2:1–2:36.
- [2012] Krötzsch, M.; Simančík, F.; and Horrocks, I. 2012. A description logic primer. *CoRR* abs/1201.4089.
- [2013] Martínez, D. C.; Wang, C.; and Hitzler, P. 2013. Towards an efficient algorithm to reason over description logics extended with nominal schemas. In Faber, W., and Lembo, D., eds., *Proc. 7th Int. Conf. on Web Reasoning and Rule Systems (RR 2013)*, volume 7994 of *LNCS*, 65–79. Springer.
- [2011] Mehdi, A., and Rudolph, S. 2011. Revisiting semantics for epistemic extensions of description logics. In Burgard, W., and Roth, D., eds., *Proc. 25th AAAI Conf. on Artificial Intelligence (AAAI'11)*. AAAI Press.
- [2005] Motik, B.; Sattler, U.; and Studer, R. 2005. Query answering for OWL DL with rules. *J. of Web Semantics* 3(1):41–60.
- [2009] Motik, B.; Shearer, R.; and Horrocks, I. 2009. Hypertableau reasoning for description logics. *J. of Artificial Intelligence Research* 36:165–228.
- [2010] Ortiz, M.; Rudolph, S.; and Simkus, M. 2010. Worst-case optimal reasoning for the Horn-DL fragments of OWL 1 and 2. In Lin, F.; Sattler, U.; and Truszczyński, M., eds., *Proc. 12th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'10)*, 269–279. AAAI Press.
- [2008a] Rudolph, S.; Krötzsch, M.; and Hitzler, P. 2008a. All elephants are bigger than all mice. In Baader, F.; Lutz, C.; and Motik, B., eds., *Proc. 21st Int. Workshop on Description Logics (DL'08)*, volume 353 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- [2008b] Rudolph, S.; Krötzsch, M.; and Hitzler, P. 2008b. Cheap Boolean role constructors for description logics. In Hölldobler, S.; Lutz, C.; and Wansing, H., eds., *Proc. 11th European Conf. on Logics in Artificial Intelligence (JELIA'08)*, volume 5293 of *LNAI*, 362–374. Springer.
- [2013] Steigmüller, A.; Glimm, B.; and Liebig, T. 2013. Nominal schema absorption. In Rossi, F., ed., *Proc. 23rd Int. Joint Conf. on Artificial Intelligence (IJCAI'13)*, 1104–1110. AAAI Press/IJCAI.
- [2000] Tobies, S. 2000. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *J. of Artificial Intelligence Research* 12:199–217.
- [2001] Tobies, S. 2001. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. Ph.D. Dissertation, RWTH Aachen, Germany.
- [27 October 2009] W3C OWL Working Group. 27 October 2009. *OWL 2 Web Ontology Language: Document Overview*. W3C Recommendation. Available at <http://www.w3.org/TR/owl2-overview/>.
- [2013] Wang, C., and Hitzler, P. 2013. A resolution procedure for description logics with nominal schemas. In Takeda, H.; Qu, Y.; Mizoguchi, R.; and Kitamura, Y., eds., *Proc. 2nd Joint Int. Conf. on Semantic Technology (JIST'12)*, volume 7774 of *LNCS*, 1–16. Springer.