Attributed Description Logics: Ontologies for Knowledge Graphs (Extended Technical Report)

Markus Krötzsch⋆, Maximilian Marx, Ana Ozaki, and Veronika Thost

Center for Advancing Electronics Dresden (cfaed), TU Dresden
`firstname.lastname@tu-dresden.de`

Abstract In modelling real-world knowledge, there often arises a need to represent and reason with meta-knowledge. To equip description logics (DLs) for dealing with such ontologies, we enrich DL concepts and roles with finite sets of attribute–value pairs, called annotations, and allow concept inclusions to express constraints on annotations. We show that this may lead to increased complexity or even undecidability, and we identify cases where this increased expressivity can be achieved without incurring increased complexity of reasoning. In particular, we describe a tractable fragment based on the lightweight description logic $\mathcal{EL}$, and we cover $\mathcal{SROIQ}$, the DL underlying OWL 2 DL.

1 Introduction

Modern data management has re-discovered the power and flexibility of graph-based representation formats, and so-called *knowledge graphs* are now used in many practical applications, e.g., in companies such as Google or Facebook. The shift towards graphs is motivated by the need for integrating knowledge from a variety of heterogeneous sources into a common format.

Description logics (DLs) seem to be an excellent fit for this scenario, since they can express complex schema information on graph-like models, while supporting incomplete information via the open world assumption. Ontology-based query answering has become an important research topic, with many recent results and implementations, and the W3C OWL and SPARQL standards provide a basis for practical adoption. One would therefore expect to encounter DLs in many applications of knowledge graphs.

However, this is not the case. While OWL is often used in RDF-based knowledge graphs developed in academia, such as DBpedia [5] and Bio2RDF [4], it has almost no impact on other applications of graph-structured data. This might in part be due to a format mismatch. Like DLs, many knowledge graphs use directed, labelled graph models, but unlike DLs they often add *(sets of) annotations* to vertices and edges. For example, the fact that Liz Taylor married Richard Burton can be described by an assertion $\mathsf{spouse}(\mathsf{taylor}, \mathsf{burton})$, but in practice we may also wish to record that they married in 1964 in Montreal, and that the marriage ended in 1974. We may write this as follows:

$$\mathsf{spouse}(\mathsf{taylor}, \mathsf{burton})@[\mathsf{start} : 1964, \mathsf{location} : \mathsf{Montreal}, \mathsf{end} : 1974] \quad (1)$$

⋆ The author thanks the competent and friendly staff of trauma surgery ward OUC-S2 at the University Hospital *Carl Gustav Carus*, Dresden, where some of this research has been executed.

Such annotated graph edges today are widespread in practice. Prominent representatives include *Property Graph*, the data model used in many graph databases [19], and *Wikidata*, the knowledge graph used by Wikipedia [23]. Looking at Wikidata as one of the few freely accessible graphs outside academia, we obtain several requirements:

– *No single purpose.* Annotations are used for many modelling tasks. Expected cases such as validity time and provenance are important, but are by far not the only uses, as (1) (taken from Wikidata) illustrates. Besides *start*, *end*, and *location*, over 150 other attributes are used at least 1000 times as annotations on Wikidata.
– *Multi-graphs.* It can be necessary to include the same assertion multiple times with different annotations. For example, Wikidata in addition to (1) also includes the assertion $\mathsf{spouse}(\mathsf{taylor}, \mathsf{burton})@[\mathsf{start}:1975, \mathsf{end}:1976]$. Such multi-graphs are also supported by Property Graph, but not by logics with functional annotations, such as semi-ring approaches [10,21] and aRDF [22].
– *Multi-attribute annotations.* Wikidata (but not Property Graph) further supports annotations where the same attribute has more than one value. Among others, Wikidata includes, e.g., the assertion $\mathsf{castMember}(\mathsf{Sesame\_Street}, \mathsf{Frank\_Oz})@[\mathsf{role}:\mathsf{Bert}, \mathsf{role}:\mathsf{Cookie\_Monster}, \mathsf{role}:\mathsf{Grover}]$.

One can encode annotated (multi-)graphs as directed graphs, e.g., using reification [9], but DLs cannot express much over such a model. For example, one cannot say that the spouse relation is symmetric, where annotations are the same in both directions [16]. Other traditional KR formalisms are similarly challenged in this situation.

In a recent work, we have therefore proposed to develop logics that support sets of attribute–value annotations natively [16]. The according generalisation of first-order logic, called *multi-attribute predicate logic* (MAPL), is expressive enough to capture weak second-order logic, making reasoning non-semi-decidable. For that reason, we have developed the Datalog-like *MAPL rule language* (MARPL) as a decidable fragment.

In this paper, we explore the use of description logics as a basis for decidable, and even tractable, fragments of MAPL. The resulting family of *attributed DLs* allows statements such as $\mathsf{spouse}@X \sqsubseteq \mathsf{spouse}^-@X$ to say that spouse is symmetric. We introduce set variables (X in the example) to refer to annotations. We refer to variables to express constraints over annotations and to compare attribute values between them. A challenge is to add functionality of this type without giving up the nature of a DL.

Another challenge is that these extensions may greatly increase the complexity of DLs. We show that reasoning becomes 2ExpTime-complete for attributed $\mathcal{ALCH}$, a prototypical DL; ExpTime-complete for attributed $\mathcal{EL}$, a DL close to OWL 2 EL; and N2ExpTime-complete for attributed $\mathcal{SROIQ}$, the DL underlying OWL 2 DL. Slight extensions of our DLs even lead to undecidability. We develop syntactic constraints to recover lower complexities, including PTime-completeness for attributed $\mathcal{EL}$.

For readability, some proofs have been moved to the appendix.

2 Attributed Description Logics

We introduce attributed description logics by defining the syntax and semantics of attributed $\mathcal{ALCH}$, denoted $\mathcal{ALCH}_{@+}$. This allows us to illustrate the central ideas without having to deal with the full generality of $\mathcal{SROIQ}$, which we introduce in Section 6. We note that fact entailment can be polynomially reduced in the DLs we study.

2.1 Syntax and Intuition

We first give the syntax and intuitive semantics of $\mathcal{ALCH}_{@+}$; the semantics will be formalised thereafter.

Example 1. We start with a guiding example, which will be formally explained when we define $\mathcal{ALCH}_{@+}$. Wikidata contains assertions of the form $\mathsf{educatedAt}(\mathsf{a\_person}, \mathsf{a\_university})@[\mathsf{start} : 2005, \mathsf{end} : 2009, \mathsf{degree} : \mathsf{master}]$. This motivates the following $\mathcal{ALCH}_{@+}$ axiom:

$$X : \lfloor \mathsf{degree} : \mathsf{master} \rfloor \quad \big(\exists \mathsf{educatedAt}@X.\mathsf{University} \quad \sqsubseteq \mathsf{MSc}@[\mathsf{start} : X.\mathsf{end}]\big) \quad (2)$$

The underlying DL axiom is $\exists\mathsf{educatedAt}.\mathsf{University} \sqsubseteq \mathsf{MSc}$, stating that anybody educated at some university holds an M.Sc. Axiom (2) restricts this to educatedAt assertions whose annotations X specify the degree to be a master, where X may contain further attribute–value pairs. Indeed, if X specifies an end date for the education, then this is used as a start for the entailed MSc assertion. Similarly, we may express that a person that was educatedAt some institution (where the degree attribute has some value) obtained a degree from this institution:

$$\mathsf{educatedAt}@\lfloor \mathsf{degree} : + \rfloor \sqsubseteq \mathsf{obtainedDegreeFrom} \quad (3)$$

Attributed DLs are defined over the usual DL signature with sets of *concept names* $\mathsf{N_C}$, *role names* $\mathsf{N_R}$, and *individual names* $\mathsf{N_I}$. In OWL terminology, concepts correspond to classes, roles correspond to properties, and individual names correspond to individuals. We consider an additional set $\mathsf{N_V}$ of *(set) variables*. Following the definition of multi-attributed predicate logic (MAPL, [16]), we define annotation sets as finite binary relations, understood as sets of attribute–value pairs. In particular, *attributes* refer to domain elements and are syntactically denoted by individual names. To describe annotation sets, we introduce *specifiers*. The set **S** of specifiers contains the following expressions:

- set variables $X \in \mathsf{N_V}$;
- *closed specifiers* $[a_1 : v_1, \ldots, a_n : v_n]$; and
- *open specifiers* $\lfloor a_1 : v_1, \ldots, a_n : v_n \rfloor$,

where $a_i \in \mathsf{N_I}$ and v_i is either $+$, an individual name in $\mathsf{N_I}$, or an expression of the form $X.c$, with X a set variable in $\mathsf{N_V}$ and c an individual name in $\mathsf{N_I}$. Intuitively, closed specifiers define specific annotation sets whereas open specifiers merely provide lower bounds. We use $+$ for "one or more" values, while $X.c$ refers to the (finite, possibly empty) set of all values of attribute c in an annotation set X. A *ground specifier* is a specifier that does not contain expressions of the form $X.c$.

Example 2. The open specifier $\lfloor \mathsf{degree} : \mathsf{master} \rfloor$ in Example 1 describes all annotation sets with at least the given attribute–value pair. The closed specifier $[\mathsf{start} : X.\mathsf{end}]$ denotes the (unique) annotation set with start as the only attribute, having exactly the values given for attribute end in X.

The set $\mathbf{R}$ of $\mathcal{ALCH}_{@+}$ *role expressions* contains all expressions $r@S$ with $r \in \mathsf{N_R}$ and $S \in \mathbf{S}$. The set $\mathbf{C}$ of $\mathcal{ALCH}_{@+}$ *concept expressions* is defined as follows

$$\mathbf{C} ::= \top \mid \bot \mid \mathsf{N_C}@\mathbf{S} \mid \neg\mathbf{C} \mid \mathbf{C} \sqcap \mathbf{C} \mid \mathbf{C} \sqcup \mathbf{C} \mid \exists\mathbf{R}.\mathbf{C} \mid \forall\mathbf{R}.\mathbf{C} \tag{4}$$

An $\mathcal{ALCH}_{@+}$ *concept* (or *role*) *assertion* is an expression $A(a)@S$ (or $r(a,b)@S$), with $A \in \mathsf{N_C}$ (or $r \in \mathsf{N_R}$), $a, b \in \mathsf{N_I}$, and $S \in \mathbf{S}$ a specifier that is not a set variable. An $\mathcal{ALCH}_{@+}$ *concept inclusion* is an expression of the form

$$X_1 : S_1, \ldots, X_n : S_n \quad (C \sqsubseteq D), \tag{5}$$

where $C, D \in \mathbf{C}$ are $\mathcal{ALCH}_{@+}$ concept expressions, $S_1, \ldots, S_n \in \mathbf{S}$ are specifiers, and $X_1, \ldots, X_n \in \mathsf{N_V}$ are set variables occurring in C, D or in $S_1, \ldots, S_n$. $\mathcal{ALCH}_{@+}$ *role inclusions* are defined analogously, but with role expressions instead of the concept expressions. An $\mathcal{ALCH}_{@+}$ *ontology* is a set of $\mathcal{ALCH}_{@+}$ assertions, and role and concept inclusions.

To simplify notation, we omit the specifier $\lfloor\rfloor$ (meaning "any annotation set") in role or concept expressions, as done for University in Example 1. In this sense, any $\mathcal{ALCH}$ axiom is also an $\mathcal{ALCH}_{@+}$ axiom. Moreover, we omit prefixes of the form $X : \lfloor\rfloor$, which merely state that X might be any annotation set.

We follow the usual DL notation for referring to other attributed DLs, where we add symbols to the DL name to indicate additional features, and remove symbols to indicate restrictions. Thus, $\mathcal{ALC}_{@+}$ denotes $\mathcal{ALCH}_{@+}$ without role hierarchies, and $\mathcal{ALCH}_{@}$ corresponds to the fragment of $\mathcal{ALCH}_{@+}$ that disallows $+$ in specifiers.

2.2 Formal Semantics

As usual in DLs, an interpretation $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$ consists of a domain $\Delta^\mathcal{I}$ and an interpretation function $\cdot^\mathcal{I}$. Individual names $c \in \mathsf{N_I}$ are interpreted as elements $c^\mathcal{I} \in \Delta^\mathcal{I}$. Concepts and roles are interpreted as relations that here include annotation sets:

- $A^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \mathcal{P}_{\mathsf{fin}}\left(\Delta^\mathcal{I} \times \Delta^\mathcal{I}\right)$ for a concept $A \in \mathsf{N_C}$, and
- $r^\mathcal{I} \subseteq (\Delta^\mathcal{I} \times \Delta^\mathcal{I}) \times \mathcal{P}_{\mathsf{fin}}\left(\Delta^\mathcal{I} \times \Delta^\mathcal{I}\right)$ for a role $r \in \mathsf{N_R}$,

where $\mathcal{P}_{\mathsf{fin}}\left(\Delta^\mathcal{I} \times \Delta^\mathcal{I}\right)$ denotes the set of all finite binary relations over $\Delta^\mathcal{I}$. Expressions with free set variables are interpreted using variable assignments $\mathcal{Z} : \mathsf{N_V} \to \mathcal{P}_{\mathsf{fin}}\left(\Delta^\mathcal{I} \times \Delta^\mathcal{I}\right)$. For an interpretation $\mathcal{I}$ and a variable assignment $\mathcal{Z}$, we define the semantics of specifiers as follows:

$$\begin{aligned}
X^{\mathcal{I},\mathcal{Z}} &:= \{\mathcal{Z}(X)\}, \\
[a : b]^{\mathcal{I},\mathcal{Z}} &:= \{\{\langle a^\mathcal{I}, b^\mathcal{I} \rangle\}\}, \\
[a : X.b]^{\mathcal{I},\mathcal{Z}} &:= \{\{\langle a^\mathcal{I}, \delta \rangle \mid \text{there is } \delta \in \Delta^\mathcal{I} \text{ such that } \langle b^\mathcal{I}, \delta \rangle \in \mathcal{Z}(X)\}\}, \\
[a : +]^{\mathcal{I},\mathcal{Z}} &:= \{\{\langle a^\mathcal{I}, \delta_1 \rangle, \ldots, \langle a^\mathcal{I}, \delta_\ell \rangle\} \mid \ell \geq 1 \text{ and } \delta_i \in \Delta^\mathcal{I}\}, \\
[a_1 : v_1, \ldots, a_n : v_n]^{\mathcal{I},\mathcal{Z}} &:= \Big\{\bigcup_{i=1}^{n} \Psi_i \,\Big|\, \Psi_i \in [a_i : v_i]^{\mathcal{I},\mathcal{Z}}\Big\}, \\
\lfloor a_1 : v_1, \ldots, a_n : v_n \rfloor^{\mathcal{I},\mathcal{Z}} &:= \Big\{\Psi \in \mathcal{P}_{\mathsf{fin}}\left(\Delta^\mathcal{I} \times \Delta^\mathcal{I}\right) \,\Big|\, \Psi \supseteq \Phi \\
&\qquad \text{for some } \Phi \in [a_1 : v_1, \ldots, a_n : v_n]^{\mathcal{I},\mathcal{Z}}\Big\},
\end{aligned}$$

where $X \in \mathsf{N_V}$, $a, a_i, b \in \mathsf{N_I}$, and v_i is $+$, an element of $\mathsf{N_I}$, or of the form $X.a$. We can now define the semantics of concept and role expressions:

$$A@S^{\mathcal{I},\mathcal{Z}} := \{\delta \in \Delta^{\mathcal{I}} \mid \langle \delta, \Psi \rangle \in A^{\mathcal{I}} \text{ for some } \Psi \in S^{\mathcal{I},\mathcal{Z}}\} \tag{6}$$

$$r@S^{\mathcal{I},\mathcal{Z}} := \{\langle \delta_1, \delta_2 \rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \langle \delta_1, \delta_2, \Psi \rangle \in r^{\mathcal{I}} \text{ for some } \Psi \in S^{\mathcal{I},\mathcal{Z}}\} \tag{7}$$

Observe that we quantify existentially over admissible annotations here ("some $\Psi \in S^{\mathcal{I},\mathcal{Z}}$"). However, variables and closed specifiers without $+$ are interpreted as singleton sets, so true existential quantification only occurs if S is an open specifier or if it contains $+$. All other DL constructs can now be defined as usual, e.g., $(C \sqcap D)^{\mathcal{I},\mathcal{Z}} = C^{\mathcal{I},\mathcal{Z}} \cap D^{\mathcal{I},\mathcal{Z}}$, $(\exists r.C)^{\mathcal{I},\mathcal{Z}} = \{\delta \mid \text{there is } \langle \delta, \epsilon \rangle \in r^{\mathcal{I},\mathcal{Z}} \text{ with } \epsilon \in C^{\mathcal{I},\mathcal{Z}}\}$, and $(\neg C)^{\mathcal{I},\mathcal{Z}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I},\mathcal{Z}}$. Note that we do not include annotations on $\top$, i.e. $\top^{\mathcal{I},\mathcal{Z}} = \Delta^{\mathcal{I}}$, and similarly for $\bot^{\mathcal{I},\mathcal{Z}} = \emptyset$.

Now $\mathcal{I}$ *satisfies* an $\mathcal{ALCH}_{@+}$ concept inclusion α of the form (5), written $\mathcal{I} \models \alpha$, if for all variable assignments $\mathcal{Z}$ such that $\mathcal{Z}(X_i) \in S_i^{\mathcal{I},\mathcal{Z}}$ for all $i \in \{1, \ldots, n\}$, we have $C^{\mathcal{I},\mathcal{Z}} \subseteq D^{\mathcal{I},\mathcal{Z}}$. Satisfaction of role inclusions is defined analogously. Moreover, $\mathcal{I}$ satisfies an $\mathcal{ALCH}_{@+}$ concept assertion $A(a)@S$ if $\langle a^{\mathcal{I}}, \Psi \rangle \in A^{\mathcal{I}}$ for some $\Psi \in S^{\mathcal{I}}$ (the latter is well-defined since S contains no variables). $\mathcal{I}$ satisfies an ontology if it satisfies all of its axioms. Based on this model theory, logical entailment is defined as usual.

Example 3. Consider the concept inclusion α of Example 1 and the interpretation $\mathcal{I}$ over domain $\Delta^{\mathcal{I}} = \{\mathsf{Mary}, \mathsf{John}, \mathsf{TUD}, \mathsf{start}, \mathsf{end}, \mathsf{2017}, \mathsf{2018}, \mathsf{master}, \mathsf{degree}\}$, given by

$$\begin{aligned}\mathsf{MSc}^{\mathcal{I}} &= \{\langle \mathsf{Mary}, \{\langle \mathsf{start}, 2016 \rangle\}\rangle, \langle \mathsf{John}, \{\langle \mathsf{start}, 2017 \rangle\}\rangle\},\\ \mathsf{educatedAt}^{\mathcal{I}} &= \{\langle \mathsf{Mary}, \mathsf{TUD}, \{\langle \mathsf{degree}, \mathsf{master} \rangle, \langle \mathsf{end}, 2016 \rangle\}\rangle,\\ &\quad \langle \mathsf{John}, \mathsf{TUD}, \{\langle \mathsf{degree}, \mathsf{master} \rangle, \langle \mathsf{end}, 2017 \rangle\}\rangle\}, \text{ and}\\ \mathsf{University}^{\mathcal{I}} &= \{\langle \mathsf{TUD}, \{\}\rangle\}.\end{aligned}$$

Then $\mathcal{I} \models \alpha$, i.e., $\mathcal{I}$ satisfies α.

3 Expressivity of Attributed Description Logics

In this section, we clarify some basic semantic properties of attributed DLs and the general relation of attributed DLs to other logical formalisms. As a first observation, we note that already $\mathcal{ALC}_{@+}$ is too expressive to be decidable:

Theorem 1. *Satisfiability of attributed DLs with $+$ is undecidable, even if the DL only supports $\sqcap$, and supports either only open specifiers or only closed specifiers.*

Proof. We reduce from the query answering problem for existential rules, i.e., first-order formulae of the form

$$\forall \boldsymbol{x}.p_1(x_1^1, \ldots, x_{\mathsf{ar}(p_1)}^1) \wedge \ldots \wedge p_n(x_1^n, \ldots, x_{\mathsf{ar}(p_n)}^n) \to \exists \boldsymbol{y}.p(z_1, \ldots, z_{\mathsf{ar}(q)}), \tag{8}$$

where the variables x_j^i occur among the universally quantified variables, i.e., $x_j^i \in \boldsymbol{x}$, and variables z_i might be universally or existentially quantified, i.e., $z_i \in \boldsymbol{x} \cup \boldsymbol{y}$. We require that each universally quantified variable occurs in some atom in the premise of the rule (safety), and that each existentially quantified variable occurs only once per rule. The latter is without loss of generality since rules that violate this restriction can be split into two rules using an auxiliary predicate. A fact is a formula of the form $q(c_1, \ldots, c_{\mathsf{ar}(q)})$ with constants c_i. Entailment of facts from given sets of facts and existential rules is known to be undecidable [3,8].

To translate an existential rule of the form (8), we consider DL concept names $P_{(i)}$ for each predicate symbol $p_{(i)}$, and individual names $a_1, \ldots, a_\ell$, where ℓ is the maximal arity of any such predicate. For each universally quantified variable x, let $\pi_x = \langle p_i, k\rangle$ be an (arbitrary but fixed) position at which x occurs, i.e., for which $x = x^i_k$. The rule can now be rewritten to the attributed DL axiom

$$X_1 : S_1, \ldots, X_n : S_n \quad (P_1@X_1 \sqcap \ldots \sqcap P_n@X_n \sqsubseteq P@T),$$

where the specifiers are defined as $S_i = [a_j : X_m.a_k \mid 1 \leq j \leq \mathsf{ar}(p_i) \text{ and } \pi_{x^i_j} = \langle p_m, k\rangle]$ and $T = [a_j : + \mid z_j \in \boldsymbol{y}] \cup [a_j : X_m.a_k \mid z_j \in \boldsymbol{x} \text{ and } \pi_{z_j} = \langle p_m, k\rangle]$ (note that we slightly abuse $\mid$ and $\cup$ here for a simpler presentation). For example, the rule $\forall xy.p_1(x, y) \wedge p_2(y, x) \to \exists z.p(x, z)$ is translated into the concept inclusion $X_1 : S_1, X_2 : S_2$ $(P_1@X_1 \sqcap P_2@X_2 \sqsubseteq P@[a_1 : X_1.a_1, a_2 : +])$, where $S_1 = [a_1 : X_1.a_1, a_2 : X_2.a_1]$ and $S_2 = [a_1 : X_2.a_1, a_2 : X_1.a_1]$. Observe that the specifier S_i for X_i may contain assignments of the form $a_j : X_i.a_j$: by our semantics, this merely states that a_j may have zero or more values. Facts of the form $q(c_1, \ldots, c_m)$ can be translated into assertions $Q(b)@[a_1 : c_1, \ldots, a_m : c_m]$ for an individual name b that is used in all such assertions.

Entailment of facts is preserved in this translation. Correctness is retained if we replace all closed by open specifiers, since the translated ontology admits a least model where all annotation sets are interpreted as the smallest possible sets. □

In Sections 4 and 5, we present two approaches for overcoming the undecidability of Theorem 1, namely to exclude $+$ from attributed DLs, and to restrict the use of expressions of the form $X.a$.

Example 4. It follows from Theorem 1 that $\mathcal{ALC}_{@+}$ ontologies may require models with annotation sets of unbounded size. To see this, consider the following ontology:

$$A(b)@\lfloor c : c\rfloor \tag{9}$$

$$A@X \sqsubseteq \exists r.A@\lfloor c : +, p : X.c, p : X.p\rfloor \tag{10}$$

$$A@X \sqcap A@\lfloor p : X.c\rfloor \sqsubseteq \bot \tag{11}$$

Axiom (9) defines an initial A member. Axiom (10) states that all A members have an r successor that is in A, annotated with some value for c ("current"), and values for p ("previous") that include all of its predecessor's c and p values. Axiom (11) requires that no individual in A may have a set of p values that include all of its c values. It is not hard to see that all models of this ontology include an infinite r-chain with arbitrarily large (but finite) A-related annotations sets.

It is interesting to discuss Theorem 1 in the context of our previous work on multi-attributed predicate logic (MAPL), which generalises first-order logic with annotation sets for arbitrary predicates. Indeed, our interpretations for attributed DLs are a special case of *multi-attributed relational structures* (MARS), though we do not make the unique name assumption here, since it is not common for the DLs we consider. Otherwise, attributed DLs are fragments of MAPL. Our notation $X.a$ is new, but it can be simulated in MAPL, e.g., by using function definitions [16].

MAPL is not semi-decidable, and we have proposed *MAPL rules* (MARPL) as a decidable fragment. MARPL supports + without restrictions, and it includes arbitrary predicate arities and more expressive specifiers (with some form of negation). In contrast, attributed DLs add the ability to quantify existentially over annotations, and therefore to derive partially specified annotation sets, which is the main reason for Theorem 1. In general, attributed DLs are based on the open world assumption, whereas MARPL could equivalently be interpreted under a closed world, least model semantics. Nevertheless, even without + the translation from the proof of Theorem 1 allows attributed DLs to capture rule languages, as the following result shows. Here, by *Datalog* we mean first-order Horn logic without existential quantifiers.

Theorem 2. *Attributed DLs can capture Datalog in the sense that every set $\mathbb{P}$ of Datalog rules and fact $q(c_1, \ldots, c_m)$ can be translated in linear time into an attributed DL ontology $KB_{\mathbb{P}}$ and assertion $Q(b)@S$, such that $\mathbb{P} \models q(c_1, \ldots, c_m)$ iff $KB_{\mathbb{P}} \models Q(b)@S$. This translation requires just $\sqcap$, no +, and either only open or only closed specifiers.*

The ability to capture Datalog reminds us of *nominal schemas*, the extension of DLs with "variable nominals" [14,15]. Indeed, this extension can also be captured in attributed DLs (we omit the details here). The converse is not true, e.g., since nominal schemas cannot encode annotation sets on role assertions. Role inclusion axioms such as $\mathsf{spouse}@X \sqsubseteq \mathsf{spouse}^{-}@X$ are therefore impossible. Another related formalism is DL-Lite$_A$, which supports (data) annotations on domain elements and pairs of domain elements [6]. This extension of DLs supports some forms of ternary relations. Nevertheless, the use case and complexity properties of DL-Lite$_A$ are different from the logics we study here, and it remains for future work to further explore attributed DL-Lite in more detail.

4 Reasoning in $\mathcal{ALCH}_@$

We first focus on $\mathcal{ALCH}_@$, for which we show reasoning to be decidable, albeit at a higher complexity. For a first positive result, we consider *ground* $\mathcal{ALCH}_@$, where ontologies do not contain any set variables. We show that we can translate any ground $\mathcal{ALCH}_@$ ontology into an equisatisfiable $\mathcal{ALCH}$ ontology by introducing fresh names for annotated concept and role names. This *renaming* is one of the key ingredients in obtaining decision procedures for attributed DLs.

Theorem 3. *Satisfiability of ground $\mathcal{ALCH}_@$ ontologies is* ExpTime*-complete.*

Proof. Hardness is immediate since $\mathcal{ALCH}_@$ generalises $\mathcal{ALCH}$. For membership, we reduce $\mathcal{ALCH}_@$ satisfiability to $\mathcal{ALCH}$ satisfiability. Given an $\mathcal{ALCH}_@$ ontology KB, let $KB^\dagger$ denote the $\mathcal{ALCH}$ ontology that is obtained by replacing each annotated concept name $A@S$ with a fresh concept name A_S, and each annotated role name $r@S$ with a fresh role name r_S, respectively. We then extend $KB^\dagger$ by all axioms

$$A_S \sqsubseteq A_T, \qquad \text{where } A_S \text{ and } A_T \text{ occur in translated axioms of } KB^\dagger\text{, and} \tag{12}$$

$$r_S \sqsubseteq r_T, \qquad \text{where } r_S \text{ and } r_T \text{ occur in translated axioms of } KB^\dagger \tag{13}$$

such that T is an open specifier, and the set of attribute–value pairs $a : b$ in S is a superset of the set of attribute–value pairs in T. We show that KB is satisfiable iff $KB^\dagger$ is satisfiable. The claim then follows from the well-known ExpTime-completeness of satisfiability checking in $\mathcal{ALCH}$. Given an $\mathcal{ALCH}_@$ model $\mathcal{I}$ of KB, we directly obtain an $\mathcal{ALCH}$ interpretation $\mathcal{J}$ over $\Delta^\mathcal{I}$ by undoing the renaming and applying $\mathcal{I}$, i.e., by mapping $A_S \in \mathsf{N_C}$ to $A@S^\mathcal{I}$, $r_S \in \mathsf{N_R}$ to $r@S^\mathcal{I}$, and $a \in \mathsf{N_I}$ to $a^\mathcal{I}$. Clearly, $\mathcal{J} \models KB^\dagger$. Conversely, given an $\mathcal{ALCH}$ model $\mathcal{J}$ of $KB^\dagger$, we construct an $\mathcal{ALCH}_@$-interpretation $\mathcal{I}$ over domain $\Delta^\mathcal{I} = \Delta^\mathcal{J} \cup \{\star\}$, where $\star$ is a fresh individual name, and define $a^\mathcal{I} := a^\mathcal{J}$ for all $a \in \mathsf{N_I}$. For a ground closed specifier $S = [a_1 : b_1, \ldots, a_n : b_n]$, we set $\Psi_S := S^\mathcal{I}$. Similarly, for a ground open specifier $S = \lfloor a_1 : b_1, \ldots, a_n : b_n \rfloor$, we define $\Psi_S := S^\mathcal{I} \cup \{\langle \star, \star \rangle\}$. Furthermore, let $A^\mathcal{I} := \{\langle a, \Psi_S \rangle \mid a \in A_S^\mathcal{J}$ for some specifier $S\}$ and $r^\mathcal{I} := \{\langle a, b, \Psi_S \rangle \mid \langle a, b \rangle \in r_S^\mathcal{J}$ for some specifier $S\}$. Then $\mathcal{I} \models KB$, where $\star$ ensures that axioms such as $\top \sqsubseteq A@\lfloor a : b \rfloor \sqcap \neg A@[a : b]$ remain satisfiable. □

The other important technique for dealing with attributed DLs is *grounding*, where we eliminate set variables from an ontology, thus transforming it into a ground ontology. As illustrated by the next result, this grounding may lead to an ontology of exponentially larger size, resulting in an increased complexity of reasoning.

Theorem 4. *Satisfiability of $\mathcal{ALCH}_@$ ontologies is in* 2ExpTime.

Proof. Let KB be an $\mathcal{ALCH}_@$ ontology, and let $\mathsf{N_I}^{KB}$ the set of individual names occurring in KB, extended by one fresh individual name x. The grounding $\mathsf{ground}(KB)$ of KB consists of all assertions in KB, together with grounded versions of inclusion axioms. Let $\mathcal{I}$ be an interpretation over domain $\Delta^\mathcal{I} = \mathsf{N_I}^{KB}$ satisfying $a^\mathcal{I} = a$ for all $a \in \mathsf{N_I}^{KB}$, and $\mathcal{Z} : \mathsf{N_V} \to \mathcal{P}_{\mathsf{fin}}\left(\Delta^\mathcal{I} \times \Delta^\mathcal{I}\right)$ be a variable assignment. Consider a concept inclusion α of the form $X_1 : S_1, \ldots, X_n : S_n\ (C \sqsubseteq D)$. We say that $\mathcal{Z}$ is *compatible with* α if $\mathcal{Z}(X_i) \in S_i^{\mathcal{I},\mathcal{Z}}$ for all $1 \leq i \leq n$. In this case, the *$\mathcal{Z}$-instance $\alpha_\mathcal{Z}$* of α is the concept inclusion $C' \sqsubseteq D'$ obtained by

- replacing each variable X_i with $[a : b \mid \langle a, b \rangle \in \mathcal{Z}(X_i)]$, and
- replacing every assignment $a : X_i.b$ occurring in some specifier by all assignments $a : c$ such that $\langle b, c \rangle \in \mathcal{Z}(X_i)$.

Then $\mathsf{ground}(KB)$ contains all $\mathcal{Z}$-instances $\alpha_\mathcal{Z}$ for all concept inclusions α in KB and all compatible variable assignments $\mathcal{Z}$; and analogous axioms for role inclusions. In general, there may be exponentially many different instances for each terminological axiom in KB, thus $\mathsf{ground}(KB)$ is of exponential size. We conclude the proof by showing that KB is satisfiable iff $\mathsf{ground}(KB)$ is satisfiable, the result then follows from Theorem 3. By construction, we have $KB \models \mathsf{ground}(KB)$, i.e., any model of KB is also a model of $\mathsf{ground}(KB)$. Conversely, let $\mathcal{I}$ be a model of $\mathsf{ground}(KB)$. Without loss of generality, assume that $x^{\mathcal{I}} \neq a^{\mathcal{I}}$ for all $a \in \mathsf{N}_{\mathsf{I}}^{KB} \setminus \{x\}$ (it suffices to add a fresh individual since x does not occur in *KB*). For an annotation set $\Psi \in \mathcal{P}_{\mathsf{fin}}\left(\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}\right)$, we define $\mathrm{rep}_x(\Psi)$ to be the annotation obtained from Ψ by replacing any individual $\delta \notin \mathcal{I}(\mathsf{N}_{\mathsf{I}}^{KB})$ in Ψ by $x^{\mathcal{I}}$. We let $\sim$ be the equivalence relation induced by $\mathrm{rep}_x(\Psi) = \mathrm{rep}_x(\Phi)$ and define an interpretation $\mathcal{J}$ over domain $\Delta^{\mathcal{J}} := \Delta^{\mathcal{I}}$, where $A^{\mathcal{J}} := \{\langle \delta, \Phi \rangle \mid \langle \delta, \Psi \rangle \in A^{\mathcal{I}} \text{ and } \Psi \sim \Phi\}$ for $A \in \mathsf{N}_{\mathsf{C}}$, $r^{\mathcal{J}} := \{\langle \delta, \epsilon, \Phi \rangle \mid \langle \delta, \epsilon, \Psi \rangle \in r^{\mathcal{I}} \text{ and } \Psi \sim \Phi\}$ for $r \in \mathsf{N}_{\mathsf{R}}$, and $a^{\mathcal{J}} := a^{\mathcal{I}}$ for all individual names $a \in \mathsf{N}_{\mathsf{I}}$. It remains to show that $\mathcal{J}$ is indeed a model of *KB*. Suppose for a contradiction that there is a concept inclusion α that is not satisfied by $\mathcal{J}$ (the case for role inclusions is analogous). Then we have some compatible variable assignment $\mathcal{Z}$ that leaves α unsatisfied. Let $\mathcal{Z}_x$ be the variable assignment $X \mapsto \mathrm{rep}_x(\mathcal{Z}(X))$ for all $X \in \mathsf{N}_{\mathsf{V}}$. Clearly, $\mathcal{Z}_x$ is also compatible with α. But now we have $C^{\mathcal{J},\mathcal{Z}} = C^{\mathcal{I},\mathcal{Z}_x}$ for all $\mathcal{ALCH}_{@}$ concepts C, yielding the contradiction $\mathcal{I} \not\models \alpha_{\mathcal{Z}_x}$. □

We regain decidability for $\mathcal{ALC}_{@+}$ by disallowing expressions of the form $X.a$.

Theorem 5. *Satisfiability of $\mathcal{ALCH}_{@+}$ ontologies without expressions of the form $X.a$ is in* 2ExpTime.

Proof. We reduce satisfiability in $\mathcal{ALCH}_{@+}$ (without expressions of the form $X.a$) to satisfiability in $\mathcal{ALCH}$, similar to the proof of Theorem 4. Consider an $\mathcal{ALCH}_{@+}$ ontology *KB* that contains the individual names $\mathsf{N}_{\mathsf{I}}^{KB}$, along with two fresh individual names x and x_+. The grounding proceeds as in the proof of Theorem 4, except that for $\mathcal{Z}$-instances $\alpha_{\mathcal{Z}}$ of concept inclusions α, we additionally replace each assignment $a : +$ occurring in some specifier by the assignment $a : x_+$. The exponentially large grounding again yields containment in 2ExpTime. From a model $\mathcal{J}$ of *KB*, we obtain a model $\mathcal{I}$ of $\mathsf{ground}(KB)$ by setting $\Delta^{\mathcal{I}} := \mathsf{N}_{\mathsf{I}}^{KB}$, $a^{\mathcal{I}} := a^{\mathcal{J}}$ for $a \in \mathsf{N}_{\mathsf{I}} \setminus \{x, x_+\}$, $x^{\mathcal{I}} := x$, $x_+^{\mathcal{I}} := x_+$, $A^{\mathcal{I}} := \{\langle \delta, \Psi \cup \Phi \rangle \mid \langle \delta, \Psi \rangle \in A^{\mathcal{J}}, \Phi \in \mathcal{P}\left(\{\langle a, x_+ \rangle \mid \langle a, b \rangle \in \Psi\}\right)\}$ for $A \in \mathsf{N}_{\mathsf{C}}$, and $r^{\mathcal{I}} := \{\langle \delta, \epsilon, \Psi \cup \Phi \rangle \mid \langle \delta, \epsilon, \Psi \rangle \in A^{\mathcal{J}}, \Phi \in \mathcal{P}\left(\{\langle a, x_+ \rangle \mid \langle a, b \rangle \in \Psi\}\right)\}$ for $r \in \mathsf{N}_{\mathsf{R}}$. Clearly, if $\mathcal{J}$ satisfies a concept inclusion in *KB*, then $\mathcal{I}$ satisfies a corresponding concept inclusion in $\mathsf{ground}(KB)$. Similarly, any concept inclusion satisfied by $\mathcal{I}$ must correspond to a concept inclusion satisfied by $\mathcal{J}$ since x_+ does not occur in *KB*. The converse direction follows immediately from the proof of Theorem 4. □

Both of these upper bounds are tight, as the next theorem shows:

Theorem 6. *Checking satisfiability of $\mathcal{ALC}_{@}$ ontologies without expressions of the form $X.a$ is* 2ExpTime*-hard.*

Proof (sketch). We reduce the word problem for exponentially space-bounded alternating Turing machines (ATMs) [7] to the entailment problem for $\mathcal{ALC}_{@}$ ontologies. We construct the tree of all configurations reachable from the initial configuration, encoding the transitions in the edges of the tree, i.e., each configuration is represented by an individual. The tape cells are represented as concepts carrying an annotation encoding the cell content and position (as a binary number). We mark the current head position with an additional concept, allowing us to copy each non-head position of the tape to successors in the configuration tree, while changing the tape cell at the head position and moving the head depending on the transition from the preceding configuration. As acceptance of a given configuration depends solely on the state and the successor configurations, we can propagate acceptance backwards from the leaves of the configuration tree to the initial configuration. □

5 Tractable Reasoning in Attributed $\mathcal{EL}$

In this section, we investigate $\mathcal{ALC}_@$ fragments based on the $\mathcal{EL}$ family of description logics. This family includes $\mathcal{EL}^{++}$, which forms the logical foundation of the OWL 2 EL profile and is widely used in applications such as in SNOMED CT [20], a clinical terminology with global scope. SNOMED CT also features a compositional syntax [1], which has recently been augmented with attribute sets allowing arbitrary concrete values. While concept expressions in either of the syntaxes can be translated into the other, $\mathcal{EL}^{++}$ provides no such attributes (i.e., concepts with attribute sets have to be represented by introducing new concept names). We can not only capture these attributes using our attribute–value sets, but also include them into the reasoning process. As a (simplified) example, the concept of a 500 mg Paracetamol tablet could be annotated with

$$\lfloor \mathsf{strengthMagnitude} : 500, \mathsf{tradeName} : \mathsf{PANADOL} \rfloor.$$

The basic logic is $\mathcal{EL}_@$, the fragment of $\mathcal{ALC}_@$ which uses only $\exists$, $\sqcap$, $\top$ and $\bot$ in concept expressions. Unfortunately, Theorem 2 shows that $\mathcal{EL}_@$ is ExpTime-complete, even with severe syntactic restrictions. To overcome this source of complexity, we impose a bound on the number of set variables per concept inclusion and exclude $X.a$:

Theorem 7. *Let $\ell \in \mathbb{N}$. Checking satisfiability of $\mathcal{EL}_@$ ontologies with at most ℓ variables per axiom, and without expressions of the form $X.a$ is* PTime*-complete.*

Proof. Hardness follows from the PTime-hardness of $\mathcal{EL}$ [2]. For membership, we polynomially reduce $\mathcal{EL}_@$ satisfiability to $\mathcal{ELH}$ satisfiability. Indeed, the grounding used in Theorem 4 can be restricted to annotation sets that are described in (ground) specifiers that are found in the ontology, since no new sets can be derived without $X.a$. The bounded number of variables then ensures that the grounding remains polynomial. Since neither grounding nor renaming introduce negation, the resulting ontology belongs to the $\mathcal{ELH}$ fragment of $\mathcal{ALCH}$. □

Observe that we can allow some uses of $X.a$, given that we obey certain restrictions:

Theorem 8. *Let $\ell, k \in \mathbb{N}$. Checking satisfiability of $\mathcal{EL}_@$ ontologies is* PTime*-complete if all of the following conditions are satisfied:*

(A) axioms contain at most ℓ variables,
(B) any closed or open specifier contains at most k expressions of the form $X.a$, and,
(C) if any specifier contains an assignment $a : X.b$, then it does not contain any other assignment for attribute a.

Proof. As in the proof of Theorem 7, we can obtain a polynomial grounding, but we may need to consider annotation sets that are not explicitly specified in the original ontology. But, due to condition (C), as the set of values for any attribute we only need to consider one of the polynomially many sets of values given explicitly through ground assignments in specifiers. Considering any combination of these value sets for any of the at most k attributes that use $X.a$ in assignments results in polynomially many annotation sets. □

We now show that violating any of these conditions makes satisfiability intractable.

Theorem 9. *Let KB be an $\mathcal{EL}_{@}$ ontology and consider conditions (A)–(C) of Theorem 8 with $\ell = 1$ and $k = 2$. Then deciding satisfiability of KB is*

(1) ExpTime*-hard if KB satisfies only conditions (B) and (C),*
(2) ExpTime*-hard if KB satisfies only conditions (A) and (C), and*
(3) PSpace*-hard if KB satisfies only conditions (A) and (B).*

It is an open question whether the PSpace bound in the third case is tight. Nevertheless, it implies intractability for this case. Finally, we show that also $\mathcal{EL}_{@^+}$ (without $X.a$) is intractable (recall that $\mathcal{EL}_{@^+}$ with $X.a$ is already undecidable by Theorem 1).

Theorem 10. *Checking satisfiability of $\mathcal{EL}_{@^+}$ ontologies without expressions of the form $X.a$ is* ExpTime*-complete.*

Proof. ExpTime-hardness follows from Theorem 9. From the proof of Theorem 5, we obtain an exponentially large grounding, which, together with the PTime complexity of $\mathcal{ELH}$, yields the ExpTime upper bound. □

6 Attributed OWL

In this section, we consider attributed DLs with further expressive features, so that in particular we can cover all of the expressivity of the OWL 2 DL ontology language [17]. The underlying DL is $\mathcal{SROIQ}_{@}$, which we introduce next by slightly extending our earlier definition of $\mathcal{ALCH}_{@}$. The set $\mathbf{R}$ of $\mathcal{SROIQ}_{@}$ *role expressions* contains all expressions $r@S$ and $r^{-}@S$ with $r \in \mathsf{N_R}$ and $S \in \mathbf{S}$. The set $\mathbf{C}$ of $\mathcal{SROIQ}_{@}$ *concept expressions* is defined as follows

$$\mathbf{C} ::= \top \mid \bot \mid \mathsf{N_C}@\mathbf{S} \mid \{\mathsf{N_I}\} \mid \neg\mathbf{C} \mid \mathbf{C} \sqcap \mathbf{C} \mid \mathbf{C} \sqcup \mathbf{C} \mid \exists\mathbf{R}.\mathbf{C} \mid \forall\mathbf{R}.\mathbf{C} \mid \leqslant n\,\mathbf{R}.\mathbf{C} \mid \geqslant n\,\mathbf{R}.\mathbf{C} \quad (14)$$

The new features are *nominals* $\{c\}$, which denote concepts containing one individual, and number restrictions $\leqslant n\,R.C$ and $\geqslant n\,R.C$, which express concepts of elements with at most/at least $n \geq 0$ R-successors in C. Note that we do not include annotations on nominals. This is no real restriction, since one can use axioms such as $\{c\} \equiv A_c@\lfloor\rfloor$ to introduce a concept name A_c that may hold such annotations. This allows us to use the same notion of interpretation as for $\mathcal{ALCH}_{@}$. Assertions, concept and role inclusions are defined as before, based on these extended sets of expressions. In addition, $\mathcal{SROIQ}_{@}$ supports complex role inclusion axioms of the form

$$X_1 : S_1, \ldots, X_n : S_n \quad (R_1 \circ \ldots \circ R_\ell \sqsubseteq T), \qquad (15)$$

where $R_i, T \in \mathbf{R}$ are $\mathcal{SROIQ}_{@}$ role expressions, $S_1, \ldots, S_n \in \mathbf{S}$ are specifiers, and $X_1, \ldots, X_n \in \mathsf{N_V}$ are set variables occurring among $R_i, T, S_1, \ldots, S_n$. A $\mathcal{SROIQ}_{@}$ *ontology* is a set of $\mathcal{SROIQ}_{@}$ assertions, and role and concept inclusions.

The semantics of these constructs and axioms is defined as usual [11], where the interpretation of roles and concepts takes annotations into account as in Section 2. For instance, we may express that any drug, such as a Paracetamol tablet, that contains at most one active ingredient and a certain amount of some such ingredient, such as 500 mg of Acetaminophen, has the same dose:

$$X : \lfloor\rfloor \ \mathsf{Drug} \sqcap \leqslant 1\,\mathsf{hasActiveIngredient}.\top \sqcap \exists \mathsf{hasActiveIngredient}@X.\top \sqsubseteq$$
$$\mathsf{Drug}@\lfloor \mathsf{strengthMagnitude} : X.\mathsf{strengthMagnitude} \rfloor$$

To ensure decidability of reasoning, $\mathcal{SROIQ}$ imposes two additional restrictions on ontologies: *simplicity* and *regularity* [11]. We adopt them to $\mathcal{SROIQ}_@$ as follows.

Simplicity is defined as in $\mathcal{SROIQ}$, ignoring the annotations. The set of *non-simple roles* $\mathsf{N}_\mathsf{R}^\mathsf{n} \subseteq \mathsf{N}_\mathsf{R}$ w.r.t. a $\mathcal{SROIQ}_@$ ontology is defined recursively: $t \in \mathsf{N}_\mathsf{R}^\mathsf{n}$ if t occurs on the right of an axiom of form (15) and either (1) $\ell > 1$ or (2) some non-simple role $s \in \mathsf{N}_\mathsf{R}^\mathsf{n}$ occurs on the left of the axiom. All other role names are *simple*. We now require that only simple roles occur in R in number restrictions $\leqslant n\, R.C$ and $\geqslant n\, R.C$.

A $\mathcal{SROIQ}_@$ ontology is *regular* if there is a strict partial order $\prec$ on the set $\mathsf{N}_\mathsf{R}^\pm = \mathsf{N}_\mathsf{R} \cup \{r^- \mid r \in \mathsf{N}_\mathsf{R}\}$, such that

(1) for all $R \in \mathsf{N}_\mathsf{R}^\pm$ and $s \in \mathsf{N}_\mathsf{R}$, we have $s \prec R$ iff $s^- \prec R$, and
(2) for all role inclusion axioms of form (15), the inclusion $R_1 \circ \ldots \circ R_\ell \sqsubseteq T$ has one of the following forms:

$$T@S \circ T@S \sqsubseteq T@S \qquad R_1 \circ \ldots \circ R_{\ell-1} \circ T@S \sqsubseteq T@S \qquad r^-@S \sqsubseteq r@S$$
$$R_1 \circ \ldots \circ R_\ell \sqsubseteq T@S \qquad T@S \circ R_2 \circ \ldots \circ R_\ell \sqsubseteq T@S$$

where $S \in \mathbf{S}$, $T \in \mathsf{N}_\mathsf{R}^\pm$, $r \in \mathsf{N}_\mathsf{R}$, and $R_1, \ldots, R_\ell \in \mathbf{R}$ are of form $R_1@S_1, \ldots, R_\ell@S_\ell$ such that $R_i \prec T$ for all $i \in \{1, \ldots, \ell\}$.

Note that we adopt the usual conditions from $\mathcal{SROIQ}$ for (inverted) role names, and further require that cases with the same role T on both sides use the same specifier S. As for $\mathcal{SROIQ}$, this condition can be verified in polynomial time by computing a minimal relation $\prec$ that satisfies the conditions and checking if it is a strict partial order.

For reasoning, the step from $\mathcal{ALCH}_@$ to $\mathcal{SROIQ}_@$ leads to several difficulties. First, nominals and cardinality restrictions may lead to the entailment of equalities $a \approx b$, which has consequences on annotation sets (e.g., $A@\lfloor c : a \rfloor \equiv A@\lfloor c : b \rfloor$ in this case). For obtaining complexity upper bounds by transformation to standard DLs as in Section 4, we need to axiomatise such relationships. Second, nominals may be used to restrict the overall size of the domain, e.g., when stating $\top \sqsubseteq \{a\}$. Besides the entailment of further equalities, this also changes the semantics of open specifiers (e.g., we obtain $A@\lfloor a : a \rfloor \sqsubseteq A@[a : a]$ in this case). As before, this requires suitable axiomatisation in $\mathcal{SROIQ}$. Either of these two effects may require exponentially many auxiliary axioms, leading to an N3ExpTime upper bound even for ground $\mathcal{SROIQ}_@$. However, we will show an N2ExpTime upper bound as for $\mathcal{SROIQ}$, which is tight.

Theorem 11. *Satisfiability of ground $\mathcal{SROIQ}_@$ ontologies is in* N2ExpTime.

To prove this theorem, we first translate ground $\mathcal{SROIQ}_@$ into an auxiliary DL, called $\mathcal{SROIQ}_\approx$, and then show how to reason in this DL by an exponential reduction to $\mathcal{C}^2$, the two-variable fragment with counting [18], which yields the desired N2ExpTime upper bound. The second part of the proof is split over several lemmas.

$\mathcal{SROIQ}_\approx$, in addition to the usual $\mathcal{SROIQ}$ axioms, supports concept inclusions of the form $a \approx b \Rightarrow C \sqsubseteq D$ and role inclusions of the form $a \approx b \Rightarrow R_1 \circ \ldots \circ R_\ell \sqsubseteq T$. An axiom $a \approx b \Rightarrow \alpha$ is satisfied by interpretation $\mathcal{I}$ if either $a^\mathcal{I} \neq b^\mathcal{I}$ or $\mathcal{I} \models \alpha$.

The translation from a ground $\mathcal{SROIQ}_@$ ontology KB to a $\mathcal{SROIQ}_\approx$ ontology $KB^\ddagger$ now proceeds as for ground $\mathcal{ALCH}_@$, by replacing annotated concept names $A@S$ by new names A_S, and likewise for roles. However, we now introduce names $A_S \in \mathsf{N_C}$ and $r_S \in \mathsf{N_R}$ for all possible open and closed ground specifiers over the set of individual names in KB, as opposed to only those occurring in KB. We then add two families of axioms for capturing the aforementioned effects. First, to handle individual equality, for each $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$, we add axioms $a \approx b \Rightarrow A_S \sqsubseteq A_T$ and $a \approx b \Rightarrow r_S \sqsubseteq r_T$ for every pair S, T of ground specifiers that are either both open or both closed, and where the sets of pairs in S and T are the same when replacing each occurrence of a by b. Second, to handle bounded domain size, we consider an individual name z not occurring in KB. Entailments of the form $z \approx a$ will be used to detect the bounded domain case. We can formalise this effect by axioms $z \approx a \Rightarrow \top \sqsubseteq \bigsqcup_{c \in \mathsf{N_I}^{KB}} \{c\}$, where $\mathsf{N_I}^{KB}$ is the set of individual names occurring in KB for all $a \in \mathsf{N_I}^{KB}$. To handle specifiers in this situation, we add axioms of the form

$$z \approx a \Rightarrow A_S \sqsubseteq \bigsqcup_{T \supseteq_c S} A_T \qquad \text{for all } A \in \mathsf{N_C} \text{ in } KB \text{ and } a \in \mathsf{N_I}^{KB} \tag{16}$$

where S is a ground open specifier and $T \supseteq_c S$ holds whenever T is a ground closed specifier that contains all attribute–value pairs in S. We would need a similar axiom as (16) for roles, but this would require disjunctions of arbitrary roles, which is not supported in $\mathcal{SROIQ}$. However, since these axioms only are necessary when all elements in the domain of interpretation are the interpretation of some individual name in $\mathsf{N_I}^{KB}$, we can instead use concept inclusions as follows:

$$z \approx a \Rightarrow \{b\} \sqcap \exists r_S.\{c\} \sqsubseteq \bigsqcup_{T \supseteq_c S} \exists r_T.\{c\} \quad \text{for all } r \in \mathsf{N_R} \text{ in } KB \text{ and } a, b, c \in \mathsf{N_I}^{KB} \tag{17}$$

where S and T are as above. Finally, as previously for $\mathcal{ALCH}_@$, we also add all axioms of the form (12) and (13). This finishes our construction of $KB^\ddagger$.

Lemma 1. *For any ground $\mathcal{SROIQ}_@$ ontology KB, the $\mathcal{SROIQ}_\approx$ ontology $KB^\ddagger$ is equisatisfiable and can be constructed in exponential time.*

The proof is analogous to the proof of Theorem 3 with one exception: when constructing models we do not introduce a fresh, unnamed domain element $\star$, but rather use $z^\mathcal{J}$ instead (which may or may not be named).

To complete the proof of Theorem 11, it remains to show that satisfiability checking for the exponentially larger $KB^\ddagger$ can still be done in nondeterministic double exponential time w.r.t. the size of KB. To this end, we can define simplicity and regularity for $\mathcal{SROIQ}_\approx$ as for $\mathcal{SROIQ}_@$, by ignoring the additional $\approx$-prefixes and disregarding any condition related to annotations. In particular, we obtain a strict partial order $\prec$, as before, and, since $KB^\ddagger$ only contains role inclusions translated directly from those in KB, it also satisfies the regularity restrictions. We define the $\circ$-*depth* of a regular $\mathcal{SROIQ}_\approx$

ontology $KB_{\approx}$ to be the maximal number k for which there is a chain of (inverted) roles $R_1 \prec R'_1 \prec \ldots \prec R_k \prec R'_k$, such that $KB_{\approx}$ contains complex role inclusions with R_i occurring as one of several roles on the left and R'_i on the right. Intuitively speaking, the $\circ$-depth bounds the number of axioms with $\circ$ along paths of $\prec$. Clearly, the $\circ$-depth of $KB^{\ddagger}$ is the same as for KB, in spite of the exponential increase in the number of axioms.

Lemma 2. *Checking satisfiability of a $\mathcal{SROIQ}_{\approx}$ ontology $KB_{\approx}$ of size s and $\circ$-depth d is possible in* NTIME $(2^{p(s \cdot 2^{q(d)})})$*, where p, q are some fixed polynomial functions.*

In particular, if an ontology is of size $O(2^n)$ but retains a $\circ$-depth in $O(n)$, then reasoning is still in N2ExpTime. To show this, we adapt the translation from $\mathcal{SROIQ}$ to $\mathcal{SHOIQ}$ as given by Kazakov [13], which is based on representing the effects of complex role inclusion axioms using concept inclusions. As a first step, one constructs, for any non-simple role expression R, a nondeterministic finite automaton $\mathcal{B}_R$ that describes the regular language of all sequences of roles that entail R [11]. We modify the known construction for $\mathcal{SROIQ}_{\approx}$ by allowing transitions in this automaton to be labelled not just by role expressions S, but also by conditional expressions $a \approx b \Rightarrow S$. The idea is that these transitions are only available if the precondition holds. By a slight adaptation of a similar observation of Horrocks and Sattler [12, Lemma 11], we obtain:

Lemma 3. *For a $\mathcal{SROIQ}_{\approx}$ ontology $KB_{\approx}$ and a role expression R, the size of $\mathcal{B}_R$ is bounded exponentially in the $\circ$-depth of $KB_{\approx}$.*

Kazakov considers a normal form of axioms, which we can construct analogously for $\mathcal{SROIQ}_{\approx}$ [13, Table 1]. We can ensure that conditions $a \approx b$ occur in concept inclusions only if they have the form $a \approx b \Rightarrow A \sqsubseteq B$ with $A, B \in \mathsf{N_C}$. The automaton $\mathcal{B}(R)$ is then used to replace every axiom of the form $A \sqsubseteq \forall R.B$ (which never has $\approx$-conditions) by the following axioms:

$$A \sqsubseteq A^R_q \qquad q \text{ starting state of } \mathcal{B}(R) \tag{18}$$

$$a \approx b \Rightarrow A^R_{q_1} \sqsubseteq \forall S.A^R_{q_2} \qquad q_1 \stackrel{a\approx b\Rightarrow S}{\rightarrow} q_2 \text{ a transition of } \mathcal{B}(R) \tag{19}$$

$$A^R_q \sqsubseteq B \qquad q \text{ a final state of } \mathcal{B}(R) \tag{20}$$

where the condition $a \approx b$ in axioms (19) can be omitted if it is not given. The resulting $\mathcal{SROIQ}_{\approx}$ ontology still contains axioms with preconditions $a \approx b$, but no more $\circ$. Every normalised $\mathcal{SROIQ}$ axiom α can be translated into a $\mathcal{C}^2$ formula $\mathsf{c2}(\alpha)$ as shown in [13, Table 1]. A $\mathcal{SROIQ}_{\approx}$ axiom of the form $a \approx b \Rightarrow \alpha$ accordingly can be translated as $(\exists^{=1}x.A_a(x) \wedge A_b(x)) \rightarrow \mathsf{c2}(\alpha)$. This completes the proof of Theorem 11.

We can lift this result to non-ground ontologies without an increase in complexity:

Theorem 12. *Satisfiability of $\mathcal{SROIQ}_{@}$ ontologies is* N2ExpTime*-complete.*

Proof. Hardness is immediate given the hardness of $\mathcal{SROIQ}$. The proof of membership uses the same grounding approach as the proof of Theorem 4, which is easily seen to be correct. This grounded ontology $\mathsf{ground}(KB)$ is exponentially larger than the input KB, but the regularity conditions for $\mathcal{SROIQ}_{@}$ ensure that it has the same (linearly bounded) $\circ$-depth. Moreover, while the transformation used for axiomatising ground

$\mathcal{SROIQ}_@$ ontologies is also exponential, it is polynomial in the number of possible ground annotation sets; this number remains single exponential w.r.t. the size of *KB*, even when considering $\mathsf{ground}(KB)$. Therefore, we find that the auxiliary $\mathcal{SROIQ}_\approx$ ontology $\mathsf{ground}(KB)^\ddagger$ is still only exponential w.r.t. *KB* while having a polynomial $\circ$-depth. The claimed complexity therefore follows from Lemma 2. □

7 Conclusion

Current graph-based knowledge representation formalisms suffer from an inability to handle meta-data in the form of sets of attribute–value pairs. These limitations show up even when dealing with purely abstract data and are orthogonal to datatype support in the formalisms. We therefore believe that KR formalisms must urgently take up the challenge of incorporating annotation structures into their expressive repertoire.

Our family of attributed description logics represents a potential solution in the context of DLs, and covers attributed $\mathcal{SROIQ}$, the DL underlying OWL 2 DL. In contrast to our recent findings on rule-based logics supporting similar annotations, attributed DLs often incur an increased reasoning complexity due to the open-world nature of DLs. We have presented a grounding-based decision procedure and identified the special cases of ground ontologies and structural restrictions on set variables, for which this overhead can be avoided. Now, more work is needed regarding practical reasoning algorithms in attributed DLs. We believe that similar approaches to those used for reasoning with nominal schemas might be effective here. Finally, there are surely further expressive mechanisms related to modelling with annotations which should be considered and investigated in future studies of the new field.

References

1. SNOMED CT Compositional Grammar Specification and Guide v2.02. IHTSDO (22 May 2015), http://doc.ihtsdo.org/download/doc_CompositionalGrammarSpecificationAndGuide_Current-en-US_INT_20150522.pdf
2. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Kaelbling, L., Saffiotti, A. (eds.) Proc. 19th Int. Joint Conf. on Artificial Intelligence (IJCAI'05). pp. 364–369. Professional Book Center (2005)
3. Beeri, C., Vardi, M.Y.: The implication problem for data dependencies. In: Even, S., Kariv, O. (eds.) Proc. 8th Colloquium on Automata, Languages and Programming (ICALP'81). LNCS, vol. 115, pp. 73–85. Springer (1981)
4. Belleau, F., Nolin, M., Tourigny, N., Rigault, P., Morissette, J.: Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. J. of Biomedical Informatics 41(5), 706–716 (2008)
5. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia – A crystallization point for the Web of Data. J. of Web Semantics 7(3), 154–165 (2009)
6. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rosati, R.: Linking data to ontologies: The description logic DL-Lite$_A$. In: Proceedings of the OWLED*06 Workshop on OWL: Experiences and Directions, Athens, Georgia, USA, November 10-11, 2006 (2006)
7. Chandra, A.K., Kozen, D.C., Stockmeyer, L.J.: Alternation. J. of the ACM 28(1), 114–133 (1981)

8. Chandra, A.K., Lewis, H.R., Makowsky, J.A.: Embedded implicational dependencies and their inference problem. In: Proc. 13th Annual ACM Symposium on Theory of Computation (STOC'81). pp. 342–354. ACM (1981)
9. Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., Vrandečić, D.: Introducing Wikidata to the linked data web. In: Proc. 13th Int. Semantic Web Conf. (ISWC'14). LNCS, vol. 8796, pp. 50–65. Springer (2014)
10. Green, T.J., Karvounarakis, G., Tannen, V.: Provenance semirings. In: Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 11-13, 2007, Beijing, China. pp. 31–40 (2007)
11. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. In: Doherty, P., Mylopoulos, J., Welty, C.A. (eds.) Proc. 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'06). pp. 57–67. AAAI Press (2006)
12. Horrocks, I., Sattler, U.: Decidability of $\mathcal{SHIQ}$ with complex role inclusion axioms. Artificial Intelligence 160(1), 79–104 (2004)
13. Kazakov, Y.: $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$. In: Brewka, G., Lang, J. (eds.) Proc. 11th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'08). pp. 274–284. AAAI Press (2008)
14. Krötzsch, M., Maier, F., Krisnadhi, A.A., Hitzler, P.: A better uncle for OWL: Nominal schemas for integrating rules and ontologies. In: Proc. 20th Int. Conf. on World Wide Web (WWW'11). pp. 645–654. ACM (2011)
15. Krötzsch, M., Rudolph, S.: Nominal schemas in description logics: Complexities clarified. In: Baral, C., De Giacomo, G., Eiter, T. (eds.) Proc. 14th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'14). pp. 308–317. AAAI Press (2014)
16. Marx, M., Krötzsch, M., Thost, V.: Logic on MARS: Ontologies for generalised property graphs. In: Proc. 26th Int. Joint Conf. on Artificial Intelligence (IJCAI'17). AAAI Press (2017), to appear; available at https://iccl.inf.tu-dresden.de/web/Inproceedings3141
17. OWL Working Group, W.: OWL 2 Web Ontology Language: Document Overview. W3C Recommendation (27 October 2009), available at http://www.w3.org/TR/owl2-overview/
18. Pratt-Hartmann, I.: Complexity of the two-variable fragment with counting quantifiers. J. of Logic, Language and Information 14, 369–395 (2005)
19. Rodriguez, M.A., Neubauer, P.: Constructions from dots and lines. Bulletin of the American Society for Information Science and Technology 36(6), 35–41 (2010)
20. Spackman, K.A., Campbell, K.E., Côté, R.A.: SNOMED RT: A reference terminology for health care. In: Masys, D.R. (ed.) Proc. 1997 AMIA Annual Fall Symposium. pp. 640–644. J. of the American Medial Informatics Association, Symposium Supplement, Hanley & Belfus (1997)
21. Straccia, U., Lopes, N., Lukacsy, G., Polleres, A.: A general framework for representing and reasoning with annotated Semantic Web data. In: Fox, M., Poole, D. (eds.) Proc. 24th AAAI Conf. on Artificial Intelligence (AAAI'10). AAAI Press (2010)
22. Udrea, O., Recupero, D.R., Subrahmanian, V.S.: Annotated RDF. ACM Trans. Comput. Logic 11(2), 10:1–10:41 (2010)
23. Vrandečić, D., Krötzsch, M.: Wikidata: A free collaborative knowledgebase. Commun. ACM 57(10) (2014)

A Proof of Theorem 6

Theorem 6. *Checking satisfiability of $\mathcal{ALC}_{@}$ ontologies without expressions of the form $X.a$ is* 2ExpTime*-hard.*

Proof. We reduce from the word problem for an exponentially space-bounded *alternating Turing machine* (ATM), which is 2ExpTime-hard [7].

An ATM is a tuple $\mathcal{M} = \langle Q, \Sigma, q_0, \Theta \rangle$, where

- $Q = Q_\exists \uplus Q_\forall$ is a finite set of states, partitioned into *existential states* $Q_\exists$ and *universal states* $Q_\forall$,
- Σ is a finite alphabet containing the *blank symbol* $\sqcup$,
- $q_0 \in Q$ is the *initial state*, and
- $\Theta \subseteq (Q \times \Sigma) \times (Q \times \Sigma) \times \{L, R\}$ is the *transition relation*.

A *configuration* of $\mathcal{M}$ is a word wqw' with $w, w' \in \Sigma^*$ and $q \in Q$, understood as the tape containing ww' (starting at the leftmost tape cell), each tape cell to the right of w' containing a blank, the head being at the leftmost position of w', and with current state q. Such a configuration is *universal* if $q \in Q_\forall$, and *existential* otherwise. *Successor configurations* are defined in terms of the transition, as it is usual [7]. A configuration α is *accepting* if either

- α is universal and each successor configuration of α is accepting, or
- α is existential and there is an accepting successor configuration of α.

In particular, universal configurations without successors are accepting, whereas existential configurations without successors are not.

A *computation* of $\mathcal{M}$ on input $w \in (\Sigma \setminus \{\sqcup\})^*$ is a sequence of successive configurations $\alpha_0, \alpha_1, \ldots$, where $\alpha_0 = q_0 w$ is the *initial configuration* for input w. Without loss of generality, we restrict ourselves to ATMs where computations on arbitrary inputs are finite [7]. $\mathcal{M}$ accepts a word w if the initial configuration is accepting.

Let $\mathcal{M}$ be such an exponentially space-bounded ATM and $w = \sigma_1 \sigma_2 \cdots \sigma_n$ an input word. Without loss of generality, we assume that $\mathcal{M}$ uses at most 2^n tape cells, and that $\mathcal{M}$ never moves to the left when the head is at the leftmost position. We construct an $\mathcal{ALC}_{@}$ ontology *KB* that entails $A(a)$ iff $\mathcal{M}$ accepts w.

We represent both configurations and the individual tape cells using individuals in *KB*, where we require that individuals representing configurations are connected to the corresponding successor configurations by roles encoding the transition. Without loss of generality, we assume that these individuals form a tree, which we call the *configuration tree*. Furthermore, each node of this tree, i.e., each configuration, is connected to 2^n individuals representing the tape cells. The main ingredients for our construction are as follows:

- an individual a denoting the root of the configuration tree;
- an individual cell carrying the contents of a tape cell;
- an auxiliary individual bit for counting;
- a concept A marking accepting configurations;
- a concept H marking the head position;

– a concept T marking tape cells, annotated with $\mathsf{cell} : \sigma$ for some $\sigma \in \Sigma$;
– concepts $B_0, \ldots, B_n$ for counting, annotated with $\mathsf{bit} : i$ for $i \in \{0, 1\}$;
– concepts S_q for all states $q \in Q$;
– roles r_θ for all transitions $\theta \in \Theta$; and
– a role tape connecting configurations to tape cells.

To improve readability, we abbreviate the binary encoding of some $i \in \{1, \ldots, 2^n\}$ by writing $C^i_{\mathbf{b}}$ as a shorthand for

$$\bigsqcap_{j=0}^{n} B_j @[\mathsf{bit} : i|^{\mathbf{b}}_j],$$

where $i|^{\mathbf{b}}_j$ denotes bit j in the binary representation of i (with bit 0 being the least significant bit). Hence, e.g., $C^3_{\mathbf{b}}$ is shorthand for

$$B_0 @[\mathsf{bit} : 1] \sqcap B_1 @[\mathsf{bit} : 1] \sqcap B_2 @[\mathsf{bit} : 0] \sqcap \cdots \sqcap B_n @[\mathsf{bit} : 0]. \tag{21}$$

Similarly, we write $C_{\mathbf{b}} @X_{\mathbf{b}}$ as shorthand for $B_0 @X_0 \sqcap \cdots \sqcap B_n @X_n$. We also write $\Omega^i_{\mathbf{b}}$ for $X_0 \mathbin{:} [\mathsf{bit} : 1], \ldots, X_{i-1} \mathbin{:} [\mathsf{bit} : 1], X_i \mathbin{:} [\mathsf{bit} : 0]$ and $\Omega^{i+1}_{\mathbf{b}} @X_{\mathbf{b}}$ for

$$\bigsqcap_{j=0}^{i-1} B_j @[\mathsf{bit} : 0] \sqcap B_i @[\mathsf{bit} : 1] \sqcap \bigsqcap_{j=i+1}^{n} B_j @X_j.$$

We begin by adding assertions to KB that encode the initial configuration of $\mathcal{M}$. We mark the root of the configuration tree with the initial state by adding $S_{q_0}(a)$ and initialise the tape cells with the input word:

$$\Big(\exists \mathsf{tape}.\big(T @[\mathsf{cell} : \sigma_0] \sqcap C^0_{\mathbf{b}} \sqcap H\big)\Big)(a), \tag{22}$$

$$\Big(\exists \mathsf{tape}.\big(T @[\mathsf{cell} : \sigma_i] \sqcap C^i_{\mathbf{b}}\big)\Big)(a) \qquad \text{for } 0 < i \leq n, \text{ and} \tag{23}$$

$$\Big(I \sqcap \exists \mathsf{tape}.\big(T @[\mathsf{cell} : \llcorner\!\lrcorner] \sqcap C^{n+1}_{\mathbf{b}}\big)\Big)(a), \tag{24}$$

where we use I as an auxiliary concept encoding that all tape cells further to the right contain blanks.

Next, we add concept inclusions to create the remaining blank tape cells, where the iteration is performed by repeatedly flipping bit values, and ensure that the counting remains unambiguous, i.e., for $0 \leq i < n$, we add:

$$\Omega^i_{\mathbf{b}}\Big(I \sqcap \exists \mathsf{tape}.\big(T @[\mathsf{cell} : \llcorner\!\lrcorner] \sqcap C_{\mathbf{b}} @X_{\mathbf{b}}\big) \sqsubseteq \exists \mathsf{tape}.\big(T @[\mathsf{cell} : \llcorner\!\lrcorner] \sqcap \Omega^{i+1}_{\mathbf{b}} @X_{\mathbf{b}}\big)\Big) \tag{25}$$

$$B_i @[\mathsf{bit} : 0] \sqcap B_i @[\mathsf{bit} : 1] \sqsubseteq \bot \tag{26}$$

Then, for each transition $\theta \in \Theta$, we make sure that tape contents are transferred to successor configurations, except for the tape cell at the head position:

$$\exists \mathsf{tape}.\big(T @Y \sqcap C_{\mathbf{b}} @X_{\mathbf{b}} \sqcap \neg H\big) \sqsubseteq \forall r_\theta.\exists \mathsf{tape}.\big(T @Y \sqcap C_{\mathbf{b}} @X_{\mathbf{b}}\big) \tag{27}$$

Now, it remains to modify the tape cell in the head position and to then move the head position in the successor configuration as appropriate for the transition, by flipping some of the bit-values to increment or decrement the position. We show the case for $\theta = \langle q, \sigma, q', \tau, R\rangle$; movement of the head to the left is handled analogously. For $0 \leq i < n$, we add the following concept inclusions:

$$\Omega^i_{\mathbf{b}}\Big(S_q \sqcap \exists\mathsf{tape}.\big(T@[\mathsf{cell} : \sigma] \sqcap C_{\mathbf{b}}@X_{\mathbf{b}} \sqcap H\big) \sqcap \exists\mathsf{tape}.\big(T@Y \sqcap \Omega^{i+1}_{\mathbf{b}}@X_{\mathbf{b}}\big)$$

$$\sqsubseteq \exists r_\theta.\Big(S_{q'} \sqcap \exists\mathsf{tape}.\big(T@[\mathsf{cell} : \tau] \sqcap C_{\mathbf{b}}@X_{\mathbf{b}}\big) \sqcap \exists\mathsf{tape}.\big(T@Y \sqcap \Omega^{i+1}_{\mathbf{b}}@X_{\mathbf{b}} \sqcap H\big)\Big)\Big)$$

The three conjuncts on either side of the concept inclusions correspond to the state, the old head position, and the new head position, respectively.

Finally, we add concept inclusions that propagate acceptance from the leaf nodes of the configuration tree backwards to the root of the tree. For existential configurations, we add $S_q \sqcap \exists r_\theta.A \sqsubseteq A$ for each $q \in Q_\exists$, whereas to handle universal configurations, we add, for each $q \in Q_\forall$, the concept inclusion

$$S_q \sqcap \exists\mathsf{tape}.\big(T@[\mathsf{cell} : \sigma] \sqsubseteq H\big) \sqcap \bigsqcap_{\substack{\theta\in\Theta \\ \theta=\langle q,\sigma,q',\tau,D\rangle}} \exists r_\theta.A \sqsubseteq A \tag{28}$$

where the conjunction may be empty if there are no suitable $\theta \in \Theta$.

An inductive argument shows that $KB \models A(a)$ iff $\mathcal{M}$ accepts w. □

B Proof of Theorem 9

Theorem 9. *Let KB be an $\mathcal{EL}_@$ ontology and consider conditions (A)–(C) of Theorem 8 with $\ell = 1$ and $k = 2$. Then deciding satisfiability of KB is*

(1) ExpTime*-hard if KB satisfies only conditions (B) and (C),*
(2) ExpTime*-hard if KB satisfies only conditions (A) and (C), and*
(3) PSpace*-hard if KB satisfies only conditions (A) and (B).*

Proof. (1) The proof uses an encoding of Datalog. For each rule, we introduce a concept inclusion: each variable x occurring in the rule is represented by a set variable X defined as $X : \lfloor\rfloor$, and Datalog atoms $p(x_{p_1}, \ldots, x_{p_{\mathsf{ar}(p)}})$ are represented by concept expressions $\exists p@X_{p_1}.\top \sqcap \cdots \sqcap \exists p@X_{p_{\mathsf{ar}(p)}}.\top$.

(2) The proof works by a modification of the Datalog encoding in the proof of Theorem 2. Instead of using one universally quantified variable for each atom in the premise of a rule, we use a single variable X defined as $X : \lfloor x_1 : X.x_1, \ldots, x_n : X.x_n\rfloor$, where $x_1, \ldots, x_n$ are the variables in the encoded rule. Datalog atoms $p(x_{p_1}, \ldots, x_{p_{\mathsf{ar}(p)}})$ can now be encoded with concept expressions $P@\lfloor a_1 : X.x_{p_1}, \ldots, a_{\mathsf{ar}(p)} : X.x_{p_{\mathsf{ar}(p)}}\rfloor$. Observe that this can be used to capture the original semantics and obeys the additional restrictions.

(3) The proof is by reduction from QBFSAT. Consider a quantified Boolean formula $\varphi = Q_1 a_1.\cdots Q_m a_m.(\varphi_1 \wedge \ldots \wedge \varphi_n)$ in prenex conjunctive normal form, where each φ_i is a disjunction $(l^i_1 \vee \ldots \vee l^i_{\ell_i})$, $a_1, \ldots, a_m$ are the variables occurring in φ, and each Q_i is either $\exists$ or $\forall$ $(1 \leq i \leq m)$. We construct an ontology KB_φ consisting of the assertion $A_0(a)$ and axioms for all $i \in \{0, \ldots, m-1\}$, where we add

$$A_i@X \sqsubseteq \exists r.A_{i+1}@\lfloor t : a_{i+1}, t : X.t, f : X.f \rfloor \tag{29}$$

$$A_i@X \sqsubseteq \exists r.A_{i+1}@\lfloor f : a_{i+1}, t : X.t, f : X.f \rfloor \tag{30}$$

if a_i is existentially quantified, and

$$A_i@X \sqsubseteq \exists r_t.A_{i+1}@\lfloor t : a_{i+1}, t : X.t, f : X.f \rfloor \tag{31}$$

$$\sqcap \exists r_f.A_{i+1}@\lfloor f : a_{i+1}, t : X.t, f : X.f \rfloor \tag{32}$$

if a_i is universally quantified. For a disjunction $\varphi_i = (l^i_1 \vee \ldots \vee l^i_{\ell_i})$, we use a^i_j to denote the variable in the literal l^i_j; we set $p^i_j := t$ if $l^i_j = a^i_j$, and $p^i_j := f$ if $l^i_j = \neg a^i_j$. Furthermore, we add axioms

$$A_m \sqsubseteq T_0, \quad T_n \sqsubseteq \mathsf{True}, \qquad A_0 \sqcap \mathsf{True} \sqsubseteq \bot, \tag{33}$$

$$T_{i-1} \sqcap A_m@\lfloor p^i_1 : a^i_1 \rfloor \sqsubseteq T_i, \quad \ldots, \quad T_{i-1} \sqcap A_m@\lfloor p^i_{\ell_i} : a^i_{\ell_i} \rfloor \sqsubseteq T_i, \tag{34}$$

where $1 \leq i \leq n$. Finally, for $i \in \{0, \ldots, m-1\}$, we add

$$A_i \sqcap \exists r.\mathsf{True} \sqsubseteq \mathsf{True} \tag{35}$$

if a_{i+1} is existentially quantified, and

$$A_i \sqcap \exists r_t.\mathsf{True} \sqcap \exists r_f.\mathsf{True} \sqsubseteq \mathsf{True} \tag{36}$$

if a_{i+1} is universally quantified. Then KB_φ is unsatisfiable if and only if φ is satisfiable.

□