Reasoning with Attributed Description Logics

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 $\mathcal{EL}$.

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 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 [4] and Bio2RDF [3], 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:

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

Such annotated graph edges today are widespread in practice. Prominent representatives include *Property Graph*, the data model used in many graph databases [15], and *Wikidata*, the knowledge graph used by Wikipedia [18]. 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 $\mathsf{spouse}(\mathsf{taylor}, \mathsf{burton})@[\mathsf{start} : 1975, \mathsf{end} : 1976]$. Property Graph also supports multi-graphs, making these models fundamentally different from logics with functional annotations, such as semi-ring approaches [9,16] and aRDF [17].
- *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 [8], 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 [14]. 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 [14]. The according generalisation of first-order logic, called *multi-attribute predicate logic* (MAPL), is expressive enough to capture weak second-order logic, making reasoning highly undecidable. We thus 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 reasoning in these DLs. We show that reasoning becomes 2ExpTime-complete for attributed $\mathcal{ALC}$, and ExpTime-complete for attributed $\mathcal{EL}$. Slight extensions of our DLs even lead to undecidability. We develop syntactic constraints to recover lower complexities, including PTime-completeness for attributed $\mathcal{EL}$.

2 Attributed Description Logics

We introduce attributed description logics by defining the syntax and semantics of attributed $\mathcal{ALCH}$, denoted $\mathcal{ALCH}_{@+}$. Adding further constructs, e.g., for defining $\mathcal{ALCHOIQ}_{@+}$, is easy given some basic familiarity with DLs [13]. We do not make the unique name assumption (UNA), as it can be enforced for individual names 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 $\mathsf{educatedAt(a\_person, a\_university)@[start : 2005, end : 2009, degree : master]}$. This motivates the following $\mathcal{ALCH}_{@+}$ TBox axiom:

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

The underlying DL axiom is $\exists\mathsf{educatedAt.University} \sqsubseteq \mathsf{MSc}$, stating that anybody educated at some university holds an M.Sc. Axiom (2) restricts this to $\mathsf{educatedAt}$ assertions whose annotations X specify the degree to be a master, where X may contain further attribute values. 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 $\mathsf{educatedAt}$ some institution (where the degree attribute has some value) obtained a degree from this institution:

$$\mathsf{educatedAt}@\lfloor \mathsf{degree} : + \rfloor \sqsubseteq \mathsf{obtainedDegreeFrom} \tag{3}$$

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* $\mathsf{N_I}$. We consider an additional set $\mathsf{N_V}$ of *(set) variables*. Following the definition of multi-attribute predicate logic (MAPL, [14]), we define annotation sets as finite binary relations (i.e., as sets of pairs of domain elements), understood as sets of attribute–value pairs. In particular, *attributes*, and in some cases also attribute *values*, are syntactically denoted by individual names. To describe annotation sets, we introduce *specifiers*. The set $\mathbf{S}$ of specifiers contains the following expressions:

- set variables $X \in \mathsf{N_V}$,
- *closed specifiers* $[a_1 : v_1, \ldots, a_n : v_n]$,
- *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 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 in $\mathsf{N_I}$. Intuitively, closed specifiers define specific, complete annotation sets whereas open specifiers merely provide lower bounds, i.e., describe potentially incomplete annotation sets. 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.

Example 2. The open specifier $\lfloor \mathsf{degree : master} \rfloor$ in Example 1 describes all annotation sets with at least the given attribute value. The closed specifier $[\mathsf{start} : X.\mathsf{end}]$ denotes the (unique) annotation set with start as the only attribute, having exactly the values as 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 $A(c)@S$ (or $r(c,d)@S$), with $A \in \mathsf{N_C}$ (or $r \in \mathsf{N_R}$), $c, d \in \mathsf{N_I}$, and $S \in \mathbf{S}$ a specifier that contains no set variables. An $\mathcal{ALCH}_{@+}$ *concept inclusion* (CI) 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 the set variables occurring in C, D or in $S_1, \ldots, S_n$. $\mathcal{ALCH}_{@+}$ *role inclusions* are defined similarly, but with role expressions instead of the concept expressions. An $\mathcal{ALCH}_{@+}$ *knowledge base* 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. Moreover, we omit prefixes of the form $X : \lfloor\rfloor$, which merely state that X might be any annotation set. In this sense, any $\mathcal{ALCH}$ axiom is also an $\mathcal{ALCH}_{@+}$ axiom.

We follow the usual DL notation for referring to other attributed DLs. For example, $\mathcal{ALC}_{@+}$ denotes $\mathcal{ALCH}_{@+}$ without role hierarchies. DL names without $+$, such as $\mathcal{ALCH}_{@}$, refer to the according attributed DL 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 $\cdot^\mathcal{I}$. Individuals $c \in \mathsf{N_I}$ 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 (i.e., the sets of admissible annotation sets) as follows:

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

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 a true existential only occurs if S is an open specifier or if it contains $+$. All other DL constructs can now be defined as usual, e.g., $(C \sqcap D)^{\mathcal{I},\mathcal{Z}} = C^{\mathcal{I},\mathcal{Z}} \cap D^{\mathcal{I},\mathcal{Z}}$, and $(\exists r.C)^{\mathcal{I},\mathcal{Z}} = \{\delta \mid \text{there is } \langle \delta_1, \delta_2 \rangle \in r^{\mathcal{I},\mathcal{Z}} \text{ with } \delta_2 \in C^{\mathcal{I},\mathcal{Z}}\}$. Note that we do not include annotations on $\top$, i.e. $\top^{\mathcal{I},\mathcal{Z}} = \Delta^{\mathcal{I}}$, and similarly for $\bot$.

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 $A(c)@S$ if $\langle c^{\mathcal{I}}, \Psi \rangle \in A^{\mathcal{I}}$ for some $\Psi \in S^{\mathcal{I}}$ (the latter is well-defined since S contains no variables). Finally, $\mathcal{I}$ satisfies a knowledge base 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 $\Delta^{\mathcal{I}} = \{\mathsf{Mary}, \mathsf{John}, \mathsf{TUD}, \mathsf{start}, \mathsf{end}, 2017, 2018, \mathsf{master}, \mathsf{degree}\}$, given by

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

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. *Deciding satisfiability in attributed DLs with* $+$ *is undecidable, even in quantifier-free attributed* $\mathcal{EL}$ *with 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

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

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 $q(c_1,\ldots,c_{\mathsf{ar}(q)})$, where c_i are constants. Entailment of facts from given sets of facts and existential rules is known to be undecidable [2,7].

To translate an existential rule of 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 into 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 $S_i = [a_j : X_m.a_k \mid \pi_{x^i_j} = \langle p_m, k\rangle]$ and $T = [a_j : + \mid z_j \in \boldsymbol{y}] \cup [a_j : X_m.a_k \mid z_j \in \boldsymbol{x}$ and $\pi_{z_j} = \langle p_m, k\rangle]$ (note that we slightly abuse $\mid$ and $\cup$ here for a simpler presentation). For example, $\forall xy.p_1(x, y) \wedge p_2(y, x) \to \exists z.p(x, z)$ would be translated to $X_1 : S_1, X_2 : S_2\ (P_1@X_1 \sqcap P_2@X_2 \sqsubseteq P@[a_1 : X_1.a_1, a_2 : +])$, 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, i.e., these assignments are trivially satisfied. 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 b that is used in all such assertions.

It is easy to see that entailment of facts is preserved in this translation. The translation remains correct if we replace all closed specifiers by open specifiers, since the translated knowledge base admits a least model where all annotation sets are interpreted to be the smallest possible set. □

We present two approaches for overcoming this undecidability, namely to exclude + from attributed DLs (Section 4), and to impose additional structural restrictions on the use of $X.a$ (Section 5).

Example 4. It follows from Theorem 1 that $\mathcal{ALC}_{@+}$ knowledge bases may require models with annotation sets of unbounded size. For a simpler example that may help to illustrate the semantics of attributed DLs, consider the following knowledge base:

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

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

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

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, ensuring that a fresh individual is used as the value for c in the successor. Axiom (11) requires that no individual in A may have a set of p values that include all of its c values, and, in particular, may not be empty. It is not hard to see that all models of this knowledge base 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 [14]. Indeed, our interpretations for attributed DLs are a special case of *multi-attributed relational structures* (MARS), though we do not make the UNA. Otherwise, attributed DLs are fragments of MAPL. Our notation $X.c$ is new, but it can be simulated in MAPL, e.g., using function definitions [14].

MAPL is highly undecidable, and *MAPL Rules* (MARPL) have been proposed as a decidable fragment [14]. 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 knowledge base $KB_{\mathbb{P}}$ 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" [10,12]. 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 potentially related formalism is DL-Lite$_A$, which supports (data) annotations on domain elements and pairs of domain elements [5]. This extension of DLs therefore supports some forms of ternary relations. While structurally and conceptually very different from our present work, it seems likely that some DL-Lite$_A$ expressions can also be captured in attributed DLs. 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 explore attributed DL-Lite in more detail.

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

We now focus on attributed DLs without +, for which we show reasoning to be decidable, albeit at a higher complexity. We then derive conditions under which the complexity remains unchanged. We establish our results for the case of $\mathcal{ALCH}_@$.

For a first positive result, we consider *ground* $\mathcal{ALCH}_@$, for which knowledge bases do not contain any set variables. We can translate any $\mathcal{ALCH}$ knowledge base into ground $\mathcal{ALCH}_@$ by replacing each concept name A and role name r by $A@\lfloor\rfloor$ and $r@\lfloor\rfloor$, respectively. Theorem 3 shows that, conversely, we can translate any ground $\mathcal{ALCH}_@$ KB into an $\mathcal{ALCH}$ KB of polynomial size.

Theorem 3. *Satisfiability of ground $\mathcal{ALCH}_@$ knowledge bases 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 $\mathcal{ALCH}_@$ KB KB, let $KB^\dagger$ denote the $\mathcal{ALCH}$ KB 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

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

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

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. □

For obtaining upper bounds in the case of general $\mathcal{ALCH}_@$, we now show that any $\mathcal{ALCH}_@$ knowledge base can be translated into a ground $\mathcal{ALCH}_@$ knowledge base, albeit at the cost of an exponential increase in size.

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

Proof. Let KB be an $\mathcal{ALCH}_@$ KB, 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 $a \in \mathsf{N_I}^{KB} \setminus \{x\}$ (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 $A^{\mathcal{J}} := \{\langle\delta, \Phi\rangle \mid \langle\delta, \Psi\rangle \in A^{\mathcal{I}}$ and $\Psi \sim \Phi\}$ for $A \in \mathsf{N}_\mathsf{C}$, $r^{\mathcal{J}} := \{\langle\delta, \epsilon, \Phi\rangle \mid \langle\delta, \epsilon, \Psi\rangle \in r^{\mathcal{I}}$ 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 $\mathcal{I} \not\models \alpha_{\mathcal{Z}_x}$. ◻

This upper bound for $\mathcal{ALCH}_@$ is tight. The full proof can be found in [11].

Theorem 5. *Satisfiability of $\mathcal{ALCH}_@$ knowledge bases is* 2ExpTime-*hard.*

Proof (sketch). We reduce the word problem for exponentially space-bounded Alternating Turing Machines (ATMs) [6] to the entailment problem for $\mathcal{ALCH}_@$ KBs. We construct the tree of all configurations reachable from the initial configuration, encoding the transitions in the edges of the tree. Individual tape cells are represented as concepts carrying an annotation encoding the cell position (as a binary number) and the cell's content. We mark the current head position with an additional concept. This allows 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 consider attributed DLs that are based on the $\mathcal{EL}$ family of description logics, and we establish various related complexity results. In particular, we study $\mathcal{EL}_@$, 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, one must impose a bound on the number of set variables per concept inclusion. In addition, we exclude $X.a$:

Theorem 6. *Let $\ell \geq 0$ be a natural number. Checking satisfiability of $\mathcal{EL}_@$ knowledge bases with at most ℓ variables per axiom, and without expressions of the form $X.a$ is* PTime-*complete.*

Proof. Hardness follows from the hardness of $\mathcal{EL}$ [1]. 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 knowledge base, since no new sets can be derived without $X.a$. The bounded number of variables then ensures that the grounding remains polynomial. ◻

It is indeed necessary to bound the number of set variables.

Theorem 7. *Satisfiability checking for $\mathcal{EL}_@$ knowledge bases without expressions of the form $X.a$ is* ExpTime*-hard.*

Proof. The proof also uses an encoding of Datalog. For each rule, we introduce a CI: each variable x occurring in the premise of the rule, is represented by a set variable X defined as $X : \lfloor\rfloor$, and the conjunction of Datalog atoms $p(x_{p_1}, \dots, x_{p_{\mathsf{ar}(p)}})$ is represented by a conjunction of corresponding concept expressions $\exists p@X_{p_1}.\top \sqcap \dots \sqcap \exists p@X_{p_{\mathsf{ar}(p)}}.\top$. It is easy to see that this encoding captures the original semantics and does not contain expressions of the form $X.a$. □

To extend Theorem 6, we can allow some uses of $X.a$ under further restrictions:

Theorem 8. *Let $\ell, k \geq 0$ be arbitrary, but fixed natural numbers. Checking satisfiability of $\mathcal{EL}_@$ knowledge bases is* PTime*-complete for KBs where*

(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 $a : v$ for the same attribute a.

Proof. As in Theorem 6, we can obtain a polynomial grounding, but we may need to consider annotation sets that are not explicitly specified in the original knowledge base. However, 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 $\leq k$ attributes that use $X.a$ in assignments results in a polynomially large set of annotation sets. □

Clearly, satisfiability of $\mathcal{ALCH}_@$ KBs obeying these conditions is ExpTime-complete.

Theorem 9. *Let KB be an $\mathcal{EL}_@$ KB 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) coNP*-hard*[1] *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 $p(x_{p_1}, \dots, x_{p_{\mathsf{ar}(p)}})$ are represented by concept expressions $\exists p@X_{p_1}.\top \sqcap \dots \sqcap \exists p@X_{p_{\mathsf{ar}(p)}}.\top$.

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

[1] This can be strengthened to PSpace-hardness [11], but here we just want to show intractability.

(3) The proof is by reduction from SAT. Let $\varphi = (\varphi_1 \wedge \ldots \wedge \varphi_n)$ be a propositional formula in conjunctive normal form, where each φ_i is a disjunction $(l^i_1 \vee \ldots \vee l^i_{\ell_i})$. Let $a_1, \ldots, a_m$ be the propositional variables in φ. We construct a knowledge base KB_φ that contains the following axioms for all $i = 0, \ldots, m-1$:

$$A_i@X \sqsubseteq \exists r.A_{i+1}@\lfloor t : a_{i+1}, t : X.t, f : X.f \rfloor \tag{14}$$

$$A_i@X \sqsubseteq \exists r.A_{i+1}@\lfloor f : a_{i+1}, t : X.t, f : X.f \rfloor \tag{15}$$

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, and we set $p^i_j := t$ if $l^i_j = a^i_j$, and $p^i_j := f$ if $l^i_j = \neg a^i_j$. Now KB_φ further contains:

$$A_m \sqsubseteq T_0 \qquad T_{i-1} \sqcap A_m@\lfloor p^i_1 : a^i_1, \ldots, p^i_{\ell_i} : a^i_{\ell_i} \rfloor \sqsubseteq T_i \qquad T_n \sqsubseteq \bot \tag{16}$$

Then KB_φ is unsatisfiable if and only if φ is satisfiable.

□

6 Conclusion

The practical impact of current knowledge representation formalisms in graph-based knowledge management is severely limited by the inability of the former to handle meta-data in the form of sets of attribute–value pairs. While some such annotations are dealing with concrete data, such as validity times, the problem is orthogonal to the logic's datatype support, and surfaces already when dealing with purely abstract data. We therefore believe that KR formalisms must urgently take up the challenge of incorporating annotation structures into their expressive repertoire.

We have presented a very first study of a potential solution for this issue in the context of description logics, thereby introducing the family of attributed DLs. Our findings differ significantly from recent results on rule-based logics that support similar annotations, since the open world assumption of DLs can make reasoning more complicated. We have identified decidable cases, for which we have developed a general grounding-based decision procedure. Two special cases, namely ground knowledge bases and structural restrictions on the use of set variables, were shown to avoid the increase in reasoning complexity that this approach may otherwise incur.

Many questions are left open in this work. The extension of our approaches to more expressive DLs with features like nominals or number restrictions is not completely trivial, since equality reasoning would require a more advanced axiomatisation during grounding. Similarly, structural restrictions of highly expressive DLs, especially RBox regularity, need to be extended to the new setting.

At the same time, more work is needed to lay the foundations of practical reasoning algorithms in attributed DLs. We believe that approaches similar to the one used for practical 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.

Acknowledgements. This work is partly supported by the German Research Foundation (DFG) in CRC 912 (HAEC) and in Emmy Noether grant KR 4381/1-1.

References

1. 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)
2. 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)
3. 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)
4. 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)
5. 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)
6. Chandra, A.K., Kozen, D.C., Stockmeyer, L.J.: Alternation. J. of the ACM 28(1), 114–133 (1981)
7. 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)
8. 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)
9. 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)
10. 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)
11. Krötzsch, M., Marx, M., Ozaki, A., Thost, V.: Attributed description logics: Ontologies for knowledge graphs. Tech. rep., TU Dresden (2017), https://iccl.inf.tu-dresden.de/web/Techreport3032/en
12. 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)
13. Krötzsch, M., Simančík, F., Horrocks, I.: Description logics. IEEE Intelligent Systems 29, 12–19 (2014)
14. 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
15. 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)
16. 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)
17. Udrea, O., Recupero, D.R., Subrahmanian, V.S.: Annotated RDF. ACM Trans. Comput. Logic 11(2), 10:1–10:41 (2010)
18. Vrandečić, D., Krötzsch, M.: Wikidata: A free collaborative knowledgebase. Commun. ACM 57(10) (2014)