Complexity of Reasoning over Entity-Relationship Models*

A. Artale[1], D. Calvanese[1], R. Kontchakov[2], V. Ryzhikov[1] and M. Zakharyaschev[2]

[1] Faculty of Computer Science
Free University of Bozen-Bolzano
I-39100 Bolzano, Italy
`lastname@inf.unibz.it`

[2] School of Comp. Science and Inf. Sys.
Birkbeck College
London WC1E 7HX, UK
`{roman,michael}@dcs.bbk.ac.uk`

Abstract. We investigate the complexity of reasoning over various fragments of the Extended Entity-Relationship (EER) language, which include different combinations of the constructors for ISA between concepts and relationships, disjointness, covering, cardinality constraints and their refinement. Specifically, we show that reasoning over EER diagrams with ISA between relationships is EXPTIME-complete even when we drop both covering and disjointness for relationships. Surprisingly, when we also drop ISA between relations, reasoning becomes NP-complete. If we further remove the possibility to express covering between entities, reasoning becomes polynomial. Our lower bound results are established by direct reductions, while the upper bounds follow from correspondences with expressive variants of the description logic *DL-Lite*. The established correspondence shows also the usefulness of *DL-Lite* as a language for reasoning over conceptual models and ontologies.

1 Introduction

Conceptual modelling formalisms, such as the Entity-Relationship model [1], are used in the phase of conceptual database design where the aim is to capture at best the semantics of the modelled application. This is achieved by expressing constraints that hold on the concepts, attributes and relations representing the domain of interest through suitable constructors provided by the conceptual modelling language. Thus, on the one hand it would be desirable to make such a language as expressive as possible in order to represent as many aspects of the modelled reality as possible. On the other hand, when using an expressive language, the designer faces the problem of understanding the complex interactions between different parts of the conceptual model under construction and the constraints therein. Such interactions may force, e.g., some class (or even all classes) in the model to become inconsistent in the sense that there cannot be any database state satisfying all constraints in which the class (respectively, all classes) is populated by at least one object. Or a class may be implied to be a subclass of another one, even if this is not explicitly asserted in the model.

* Authors partially supported by the U.K. EPSRC grant GR/S63175, Tones, KnowledgeWeb and InterOp EU projects, and by the PRIN project funded by MIUR.

To understand the consequences, both explicit and implicit, of the constraints in the conceptual model being constructed, it is thus essential to provide for an automated reasoning support.

In this paper, we address these issues and investigate the complexity of reasoning in conceptual modelling languages equipped with various forms of constraints. We carry out our analysis in the context of the Extended Entity-Relationship (EER) language [2], where the domain of interest is represented via *entities* (representing sets of objects), possibly equipped with *attributes*, and *relationships* (representing relations over objects)[1]. Specifically, the kind of constraints that will be taken into account in this paper are the ones typically used in conceptual modelling, namely:

- *is-a* relations between both entities and relationships;
- *disjointness* and *covering* (referred to as the *Boolean* constructors in what follows) between both entities and relationships;
- *cardinality* constraints for participation of entities in relationships;
- *refinement* of cardinalities for sub-entities participating in relationships; and
- *multiplicity* constraints for attributes.

The hierarchy of EER languages we consider here is shown in the table below together with the complexity results for reasoning in these languages (all our languages include cardinality, refinement and multiplicity constraints).

lang.	entities			relationships			complexity
	ISA	disjoint	covering	ISA	disjoint	covering	
	$C_1 \sqsubseteq C_2$	$C_1 \sqcap C_2 \sqsubseteq \bot$	$C = C_1 \sqcup C_2$	$R_1 \sqsubseteq R_2$	$R_1 \sqcap R_2 \sqsubseteq \bot$	$R = R_1 \sqcup R_2$	
ER_{full}	+	+	+	+	+	+	ExpTime [3]
ER_{isaR}	+	+	+	+	−	−	ExpTime
ER_{bool}	+	+	+	−	−	−	NP
ER_{ref}	+	+	−	−	−	−	NLogSpace

According to [3] reasoning over UML class diagrams is ExpTime-complete, and it is easy to see that the same holds for ER_{full} diagrams as well (cf. e.g., [4]). Here we strengthen this result by showing (using reification) that reasoning is still ExpTime-complete for its sublanguage ER_{isaR}. The NP upper bound for ER_{bool} is proved by embedding ER_{bool} into *DL-Lite*$_{bool}$, the *Boolean extension* of the tractable DL *DL-Lite* [5, 6]. Thus, quite surprisingly, ISA between relationships alone is a major source of complexity of reasoning over conceptual schemas. Finally, we show that ER_{ref} is closely related to *DL-Lite*$_{krom}$, the Krom fragment of *DL-Lite*$_{bool}$, and that reasoning in it is polynomial. The correspondence between modelling languages like ER_{bool} and DLs like *DL-Lite*$_{bool}$ shows that the *DL-Lite* family are useful languages for reasoning over conceptual models and ontologies, even though they are not equipped with all the constructors that are typical of rich ontology languages such as OWL and its variants [7].

Our analysis is in spirit similar to [8], where the consistency checking problem for an EER model equipped with forms of inclusion and disjointness constraints is studied and a polynomial-time algorithm for the problem is given (assuming constant arities of relationships). Such a polynomial-time result is incomparable

[1] Our results can be adapted to other modelling formalisms, such as UML diagrams.

with the one for ER_{ref}, since ER_{ref} lacks both ISA and disjointness for relationships (both present in [8]); on the other hand, it is equipped with cardinality and multiplicity constraints. We also mention [9], where reasoning over cardinality constraints in the basic ER model is investigated and a polynomial-time algorithm for strong schema consistency is given, and [10], where the study is extended to the case where ISA between entities is also allowed and an exponential algorithm for entity consistency is provided. Note, however, that in [9, 10] the reasoning problem is analysed under the assumption that databases are finite, whereas we do not require finiteness in this paper.

2 The *DL-Lite* Language

We consider the extension $DL\text{-}Lite_{bool}$ [6] of the description logic *DL-Lite* [11, 5]. The language of $DL\text{-}Lite_{bool}$ contains *concept names* $A_0, A_1, \dots$ and *role names* $P_0, P_1, \dots$. Complex *roles* R and *concepts* C of $DL\text{-}Lite_{bool}$ are defined as follows:

$$\begin{array}{lcl} R & ::= & P_i \quad | \quad P_i^-, \\ B & ::= & \bot \quad | \quad A_i \quad | \quad \geq q\, R, \\ C & ::= & B \quad | \quad \neg C \quad | \quad C_1 \sqcap C_2, \end{array}$$

where $q \geq 1$. Concepts of the form B are called *basic concepts*. A $DL\text{-}Lite_{bool}$ *knowledge base* is a finite set of axioms of the form $C_1 \sqsubseteq C_2$. A $DL\text{-}Lite_{bool}$ *interpretation* $\mathcal{I}$ is a structure $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}} \neq \emptyset$ and $\cdot^{\mathcal{I}}$ is a function such that $A_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, for all A_i, and $P_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, for all P_i. The role and concept constructors are interpreted in $\mathcal{I}$ as usual. We also make use of the standard abbreviations: $\top := \neg\bot$, $\exists R := (\geq 1\, R)$ and $\leq q\, R := \neg(\geq q+1\, R)$. We say that $\mathcal{I}$ *satisfies* an axiom $C_1 \sqsubseteq C_2$ if $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$. A knowledge base $\mathcal{K}$ is *satisfiable* if there is an interpretation $\mathcal{I}$ that satisfies all the axioms of $\mathcal{K}$ (such an $\mathcal{I}$ is called a *model* of $\mathcal{K}$). A concept C is *satisfiable w.r.t.* $\mathcal{K}$ if there is a model $\mathcal{I}$ of $\mathcal{K}$ such that $C^{\mathcal{I}} \neq \emptyset$.

We also consider a sub-language $DL\text{-}Lite_{krom}$ of $DL\text{-}Lite_{bool}$, called the *Krom fragment*, where only axioms of the following form are allowed (with B_i basic concepts):

$$B_1 \sqsubseteq B_2, \qquad B_1 \sqsubseteq \neg B_2, \qquad \neg B_1 \sqsubseteq B_2,$$

Theorem 1 ([6]). *Concept and KB satisfiability are* NP*-complete for* $DL\text{-}Lite_{bool}$ *KBs and* NLogSpace*-complete for* $DL\text{-}Lite_{krom}$ *KBs.*

3 The Conceptual Modelling Language

In this section, we define the notion of a *conceptual schema* by providing its syntax and semantics for the fully-fledged conceptual modelling language ER_{full}. First citizens of a conceptual schema are *entities*, *relationships* and *attributes*. Arguments of relationships—specifying the role played by an entity when participating in a particular relationship—are called *roles*. Given a conceptual schema, we make the following assumptions: relationship and entity names are unique; attribute names are local to entities (i.e., the same attribute may be used by different entities; its type, however, must be the same); role names are local to relationships (this freedom will be limited when considering conceptual models without sub-relationships).

Given a finite set $X = \{x_1, \ldots, x_n\}$ and a set Y, an X-*labelled tuple over* Y is a (total) function $T\colon X \to Y$. The element $T[x] \in Y$ is said to be *labelled* by x; we also write $(x, y) \in T$ if $y = T[x]$. The set of all X-labelled tuples over Y is denoted by $T_Y(X)$. For $y_1, \ldots, y_n \in Y$, the expression $\langle x_1\colon y_1, \ldots, x_n\colon y_n\rangle$ denotes $T \in T_Y(X)$ such that $T[x_i] = y_i$, for $1 \leq i \leq n$.

Definition 1 (ER_{full} **syntax).** An ER_{full} *conceptual schema* Σ is a tuple of the form $(\mathcal{L}, \textsc{rel}, \textsc{att}, \textsc{card}_R, \textsc{card}_A, \textsc{ref}, \textsc{isa}, \textsc{disj}, \textsc{cov})$, where

- $\mathcal{L}$ is the disjoint union of alphabets $\mathcal{E}$ of *entity* symbols, $\mathcal{A}$ of *attribute* symbols, $\mathcal{R}$ of *relationship* symbols, $\mathcal{U}$ of *role* symbols and $\mathcal{D}$ of *domain* symbols; the tuple $(\mathcal{E}, \mathcal{A}, \mathcal{R}, \mathcal{U}, \mathcal{D})$ is called the *signature* of the schema Σ.
- $\textsc{rel}$ is a function assigning to every relationship symbol $R \in \mathcal{R}$ a tuple $\textsc{rel}(R) = \langle U_1\colon E_1, \ldots, U_m\colon E_m\rangle$ over the entity symbols $\mathcal{E}$ labelled with a non-empty set $\{U_1, \ldots, U_m\}$ of role symbols; m is called the *arity* of R.
- $\textsc{att}$ is a function that assigns to every entity symbol $E \in \mathcal{E}$ a tuple $\textsc{att}(E)$, $\textsc{att}(E) = \langle A_1\colon D_1, \ldots, A_h\colon D_h\rangle$, over the domain symbols $\mathcal{D}$ labelled with some (possibly empty) set $\{A_1, \ldots, A_h\}$ of attribute symbols.
- $\textsc{card}_R\colon \mathcal{R} \times \mathcal{U} \times \mathcal{E} \to \mathbb{N} \times (\mathbb{N} \cup \{\infty\})$ is a *partial* function (called *cardinality constraints*); $\textsc{card}_R(R, U, E)$ may be defined only if $(U, E) \in \textsc{rel}(R)$.
- $\textsc{card}_A\colon \mathcal{A} \times \mathcal{E} \to \mathbb{N} \times (\mathbb{N} \cup \{\infty\})$ is a *partial* function (called *multiplicity of attributes*); $\textsc{card}_A(A, E)$ may be defined only if $(A, D) \in \textsc{att}(E)$, for some $D \in \mathcal{D}$.
- $\textsc{ref}\colon \mathcal{R} \times \mathcal{U} \times \mathcal{E} \to \mathbb{N} \times (\mathbb{N} \cup \{\infty\})$ is a *partial* function (called *refinement* of cardinality constraints); $\textsc{ref}(R, U, E)$ may be defined only if $E \;\textsc{isa}\; E'$ and $(U, E') \in \textsc{rel}(R)$; note that $\textsc{ref}$ subsumes cardinality constraints $\textsc{card}_R$.
- $\textsc{isa} = \textsc{isa}_E \cup \textsc{isa}_R$, where $\textsc{isa}_R \subseteq \mathcal{E} \times \mathcal{E}$ and $\textsc{isa}_R \subseteq \mathcal{R} \times \mathcal{R}$.
- $\textsc{disj} = \textsc{disj}_E \cup \textsc{disj}_R$ and $\textsc{cov} = \textsc{cov}_E \cup \textsc{cov}_R$, where $\textsc{disj}_E, \textsc{cov}_E \subseteq 2^{\mathcal{E}} \times \mathcal{E}$ and $\textsc{disj}_R, \textsc{cov}_R \subseteq 2^{\mathcal{R}} \times \mathcal{R}$.

$\textsc{isa}_R$, $\textsc{disj}_R$ and $\textsc{cov}_R$ may only be defined for relationships of the same arity. In what follows we also use infix notation for relations $\textsc{isa}$, $\textsc{isa}_E$, etc.

Definition 2 (ER_{full} **semantics).** Let Σ be an ER_{full} conceptual schema and B_D, for $D \in \mathcal{D}$, a collection of disjoint countable sets called *basic domains*. An *interpretation* of Σ is a pair $\mathcal{B} = (\Delta^{\mathcal{B}} \cup \Lambda^{\mathcal{B}}, \cdot^{\mathcal{B}})$, where $\Delta^{\mathcal{B}} \neq \emptyset$ is the *interpretation domain*; $\Lambda^{\mathcal{B}} = \bigcup_{D \in \mathcal{D}} \Lambda^{\mathcal{B}}_D$, with $\Lambda^{\mathcal{B}}_D \subseteq B_D$ for each $D \in \mathcal{D}$, is the *active domain* such that $\Delta^{\mathcal{B}} \cap \Lambda^{\mathcal{B}} = \emptyset$; $\cdot^{\mathcal{B}}$ is a function such that $E^{\mathcal{B}} \subseteq \Delta^{\mathcal{B}}$, for each $E \in \mathcal{E}$, $A^{\mathcal{B}} \subseteq \Delta^{\mathcal{B}} \times \Lambda^{\mathcal{B}}$, for each $A \in \mathcal{A}$, $R^{\mathcal{B}} \subseteq T_{\Delta^{\mathcal{B}}}(\mathcal{U})$, for each $R \in \mathcal{R}$; and $D^{\mathcal{B}} = \Lambda^{\mathcal{B}}_D$, for each $D \in \mathcal{D}$. An interpretation $\mathcal{B}$ of Σ is called a *legal database state* if the following holds:

1. for each $R \in \mathcal{R}$ with $\textsc{rel}(R) = \langle U_1\colon E_1, \ldots, U_m\colon E_m\rangle$ and each $1 \leq i \leq m$,
   - for all $r \in R^{\mathcal{B}}$, $r = \langle U_1\colon e_1, \ldots, U_m\colon e_m\rangle$ and $e_i \in E_i^{\mathcal{B}}$;
   - if $\textsc{card}_R(R, U_i, E_i) = (\alpha, \beta)$ then $\alpha \leq \sharp\{r \in R^{\mathcal{B}} \mid (U_i, e_i) \in r\} \leq \beta$, for all $e_i \in E_i^{\mathcal{B}}$;

– if $\text{REF}(R, U_i, E) = (\alpha, \beta)$, for $E \in \mathcal{E}$ with E ISA E_i, then, for all $e \in E^{\mathcal{B}}$, $\alpha \leq \sharp\{r \in R^{\mathcal{B}} \mid (U_i, e) \in r\} \leq \beta$;

2. for each $E \in \mathcal{E}$ with $\text{ATT}(E) = \langle A_1 \colon D_1, \ldots, A_h \colon D_h\rangle$ and each $1 \leq i \leq h$,
   – for all $(e, a) \in \Delta^{\mathcal{B}} \times \Lambda^{\mathcal{B}}$, if $(e, a) \in A_i^{\mathcal{B}}$ then $a \in D_i^{\mathcal{B}}$;
   – if $\text{CARD}_A(A_i, E) = (\alpha, \beta)$ then $\alpha \leq \sharp\{(e, a) \in A_i^{\mathcal{B}}\} \leq \beta$, for all $e \in E^{\mathcal{B}}$;
3. for all $E_1, E_2 \in \mathcal{E}$, if E_1 ISA_E E_2 then $E_1^{\mathcal{B}} \subseteq E_2^{\mathcal{B}}$ (similarly for relationships);
4. for all $E, E_1, \ldots, E_n \in \mathcal{E}$, if $\{E_1, \ldots, E_n\}$ DISJ_E E then $E_i^{\mathcal{B}} \subseteq E^{\mathcal{B}}$, for every $1 \leq i \leq n$, and $E_i^{\mathcal{B}} \cap E_j^{\mathcal{B}} = \emptyset$, for $1 \leq i < j \leq n$ (similarly for relationships);
5. for all $E, E_1, \ldots, E_n \in \mathcal{E}$, $\{E_1, \ldots, E_n\}$ COV_E E implies $E^{\mathcal{B}} = \bigcup_{i=1}^{n} E_i^{\mathcal{B}}$ (similarly for relationships).

Reasoning tasks over conceptual schemas include verifying whether an entity, a relationship, or a schema is *consistent*, or checking whether an entity (or a relationship) *subsumes* another entity (relationship, respectively):

Definition 3 (Reasoning services). Let Σ be an ER_{full} schema.

- Σ is *consistent* (*strongly consistent*) if there exists a legal database state $\mathcal{B}$ for Σ such that $E^{\mathcal{B}} \neq \emptyset$, for some (*every*, respectively) entity $E \in \mathcal{E}$.
- An entity $E \in \mathcal{E}$ (relationship $R \in \mathcal{R}$) is *consistent w.r.t.* Σ if there exists a legal database state $\mathcal{B}$ for Σ such that $E^{\mathcal{B}} \neq \emptyset$ ($R^{\mathcal{B}} \neq \emptyset$, respectively).
- An entity $E_1 \in \mathcal{E}$ (relationship $R_1 \in \mathcal{R}$) *subsumes* an entity $E_2 \in \mathcal{E}$ (relationship $R_2 \in \mathcal{R}$) w.r.t. Σ if $E_2^{\mathcal{B}} \subseteq E_1^{\mathcal{B}}$ ($R_2^{\mathcal{B}} \subseteq R_1^{\mathcal{B}}$, respectively), for every legal database state $\mathcal{B}$ for Σ.

One can show that the reasoning tasks of schema/entity/relationship consistency and entity subsumption are reducible to each other. (Note that in the absence of the covering constructor schema consistency cannot be reduced to a single instance of entity consistency, though it can be reduced to several entity consistency checks.) Due to these equivalences, in the following we will consider entity consistency as the main reasoning service.

4 Complexity of Reasoning in EER Languages

This section shows the complexity results obtained in this paper for reasoning over different EER languages (All proofs can be found at `http://www.inf.unibz.it/~artale/papers/dl07-full.pdf`.)

Reasoning over ER_{isaR} schemas. The modelling language ER_{isaR} is the subset of ER_{full} without the Booleans between relationships (i.e., $\text{DISJ}_R = \emptyset$ and $\text{COV}_R = \emptyset$) but with the possibility to express ISA between them. We establish an ExpTime lower bound for satisfiability of ER_{isaR} conceptual schemas by reduction of the satisfiability problem for $\mathcal{ALC}$ knowledge bases. It is easy to show (see, e.g., [3, Lemma 5.1]) that one can convert, in a satisfiability preserving way, an $\mathcal{ALC}$ KB $\mathcal{K}$ into a *primitive* KB $\mathcal{K}'$ that contains only axioms of the form: $A \sqsubseteq B, A \sqsubseteq \neg B, A \sqsubseteq B \sqcup B', A \sqsubseteq \forall R.B, A \sqsubseteq \exists R.B$, where A, B, B' are concept

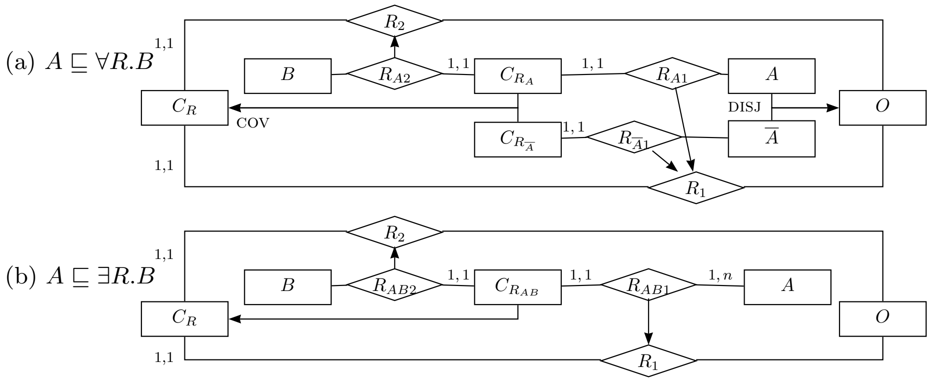

Fig. 1. Encoding axioms: (a) $A \sqsubseteq \forall R.B$; (b) $A \sqsubseteq \exists R.B$.

names and R is a role name, and the size of $\mathcal{K}'$ is linear in the size of $\mathcal{K}$. Thus, satisfiability problem for primitive $\mathcal{ALC}$ KBs is ExpTime-complete [3].

Let $\mathcal{K}$ be a primitive $\mathcal{ALC}$ KB. The reduction in [3] maps $\mathcal{K}$ into an UML class diagram. We show how to define an ER_{isaR} schema $\Sigma(\mathcal{K})$: the first three types of axioms are dealt with in a way similar to [3]. Axioms of the form $A \sqsubseteq \forall R.B$ are encoded in [3] using both the Booleans and ISA between relationships, which are unavailable in ER_{isaR}. In order to to stay within ER_{isaR}, we propose to use reification of $\mathcal{ALC}$ roles (which are binary relationships) to encode the last two types of axioms. This approach is illustrated in Fig. 1: in (a), $A \sqsubseteq \forall R.B$ is encoded by reifying the binary relationship R with the entity C_R so that the functional relationships R_1 and R_2 give the first and second component of the reified R, respectively; a similar encoding is used to capture $A \sqsubseteq \exists R.B$ in (b).

Lemma 1. *A concept name A is satisfiable w.r.t a primitive $\mathcal{ALC}$ KB $\mathcal{K}$ iff the entity A is consistent w.r.t the ER_{isaR} schema $\Sigma(\mathcal{K})$.*

Theorem 2. *Reasoning over ER_{isaR} schemas is* ExpTime-*complete.*

The lower bound follows, by Lemma 1, from ExpTime-completeness of concept satisfiability w.r.t. primitive $\mathcal{ALC}$ KBs [3] and the upper bound from the respective upper bound for ER_{full} [3].

Reasoning over ER_{bool} schemas. Denote by ER_{bool} the sub-language of ER_{full} without ISA and the Booleans between relationships (i.e., $\text{ISA}_R = \emptyset$, $\text{DISJ}_R = \emptyset$ and $\text{COV}_R = \emptyset$). In ER_{bool} we impose an insignificant syntactic restriction on REL: there is no $U \in \mathcal{U}$ such that $(U, E_i) \in \text{REL}(R_i)$, $i = 1, 2$, for some $E_1, E_2 \in \mathcal{E}$ and some *distinct* $R_1, R_2 \in \mathcal{R}$.

We define a polynomial translation τ of ER_{bool} schemas into *DL-Lite$_{bool}$* KBs. Let Σ be an ER_{bool} schema. For every entity, domain or relationship symbol $N \in \mathcal{E} \cup \mathcal{D} \cup \mathcal{R}$, we fix a *DL-Lite$_{bool}$* concept name $\overline{N}$; for every attribute or role symbol $N \in \mathcal{A} \cup \mathcal{U}$, we fix a *DL-Lite$_{bool}$* role name $\overline{N}$. The translation $\tau(\Sigma)$ of

Σ is defined as follows:

$$\tau(\Sigma) \quad = \quad \tau_{dom} \quad \cup \quad \bigcup_{R\in\mathcal{R}} \left[\tau^{R}_{rel} \cup \tau^{R}_{card_R} \cup \tau^{R}_{ref}\right] \quad \cup \quad \bigcup_{E\in\mathcal{E}} \left[\tau^{E}_{att} \cup \tau^{E}_{card_A}\right] \quad \cup$$

$$\bigcup_{\substack{E_1,E_2\in\mathcal{E}\\ E_1 \text{ISA} E_2}} \tau^{E_1,E_2}_{isa} \quad \cup \quad \bigcup_{\substack{E_1,\dots,E_n,E\in\mathcal{E}\\ \{E_1,\dots,E_n\}\text{DISJ}E}} \tau^{\{E_1,\dots,E_n\},E}_{disj} \quad \cup \quad \bigcup_{\substack{E_1,\dots,E_n,E\in\mathcal{E}\\ \{E_1,\dots,E_n\}\text{COV}E}} \tau^{\{E_1,\dots,E_n\},E}_{cov},$$

where

- $\tau_{dom} = \big\{\overline{D} \sqsubseteq \neg\overline{X} \mid D \in \mathcal{D},\ X \in \mathcal{E}\cup\mathcal{R}\cup\mathcal{D},\ D \neq X\big\}$;
- $\tau^{R}_{rel} = \big\{\overline{R} \sqsubseteq \exists\overline{U},\ \ \geq 2\,\overline{U} \sqsubseteq \bot,\ \ \exists\overline{U} \sqsubseteq \overline{R},\ \ \exists\overline{U}^{-} \sqsubseteq \overline{E} \mid (U,E) \in \text{REL}(R)\big\}$;
- $\tau^{R}_{card_R} = \big\{\overline{E}\ \sqsubseteq\ \geq \alpha\,\overline{U}^{-} \mid (U,E) \in \text{REL}(R), \text{CARD}_R(R,U,E) = (\alpha,\beta), \alpha \neq 0\big\}$
  $\cup\big\{\overline{E}\ \sqsubseteq\ \leq \beta\,\overline{U}^{-} \mid (U,E) \in \text{REL}(R), \text{CARD}_R(R,U,E) = (\alpha,\beta), \beta \neq \infty\big\}$;
- $\tau^{R}_{ref} = \big\{\overline{E}\ \sqsubseteq\ \geq \alpha\,\overline{U}^{-} \mid (U,E) \in \text{REL}(R), \text{REF}(R,U,E) = (\alpha,\beta), \alpha \neq 0\big\}$
  $\cup\big\{\overline{E}\ \sqsubseteq\ \leq \beta\,\overline{U}^{-} \mid (U,E) \in \text{REL}(R), \text{REF}(R,U,E) = (\alpha,\beta), \beta \neq \infty\big\}$;
- $\tau^{E}_{att} = \big\{\exists\overline{A}^{-} \sqsubseteq \overline{D} \mid (A,D) \in \text{ATT}(E)\big\}$;
- $\tau^{E}_{card_A} = \big\{\overline{E}\ \sqsubseteq\ \geq \alpha\,\overline{A} \mid (A,D) \in \text{ATT}(E), \text{CARD}_A(A,E) = (\alpha,\beta), \alpha \neq 0\big\}$
  $\cup\big\{\overline{E}\ \sqsubseteq\ \leq \beta\,\overline{A} \mid (A,D) \in \text{ATT}(E), \text{CARD}_A(A,E) = (\alpha,\beta), \beta \neq \infty\big\}$;
- $\tau^{E_1,E_2}_{isa} = \big\{\overline{E_1} \sqsubseteq \overline{E_2}\big\}$;
- $\tau^{\{E_1,\dots,E_n\},E}_{disj} = \big\{\overline{E_i} \sqsubseteq \overline{E} \mid 1 \leq i \leq n\big\} \cup \big\{\overline{E_i} \sqsubseteq \neg\overline{E_j} \mid 1 \leq i < j \leq n\big\}$;
- $\tau^{\{E_1,\dots,E_n\},E}_{cov} = \big\{\overline{E_i} \sqsubseteq \overline{E} \mid 1 \leq i \leq n\big\} \cup \big\{\ \overline{E} \sqsubseteq \overline{E_1} \sqcup \cdots \sqcup \overline{E_n}\big\}$.

Clearly, the size of $\tau(\Sigma)$ is polynomial in the size of Σ.

Lemma 2. *An entity E is consistent w.r.t. an ER_{bool} schema Σ iff the concept $\overline{E}$ is satisfiable w.r.t. the DL-Lite$_{bool}$ KB $\tau(\Sigma)$.*

Theorem 3. *Reasoning over ER_{bool} conceptual schemas is* NP*-complete.*

The upper bound is proved by Lemma 2 and Theorem 1; the lower one is by reduction of the NP-complete 3SAT problem to entity consistency for ER_{bool} schemas.

Reasoning over ER_{ref} schemas. Denote by ER_{ref} the modelling language without the Booleans and ISA between relationships, but with the possibility to express ISA and disjointness between entities (i.e., $\text{DISJ}_R = \emptyset$, $\text{COV}_R = \emptyset$, $\text{ISA}_R = \emptyset$ and $\text{COV}_E = \emptyset$). Thus, ER_{ref} is essentially ER_{bool} without covering.

Theorem 4. *The entity consistency problem for ER_{ref} is* NLogSpace*-complete.*

The upper bound follows from the fact that for any ER_{ref} schema, Σ, $\tau(\Sigma)$ is a *DL-Lite$_{krom}$* KB ($\tau_{cov} = \emptyset$). Thus, by Lemma 2, the entity consistency problem for ER_{ref} can be reduced to concept satisfiability for *DL-Lite$_{krom}$* KBs, which is NLogSpace-complete (see Theorem 1), while the reduction can be proved to be computed in logspace. The lower bound is obtained by reduction of the non-reachability problem in oriented graphs (the non-reachability problem is known to be coNLogSpace-complete and so, it is NLogSpace-complete as these classes coincide by the Immerman-Szelepcsényi theorem; see, e.g., [12]).

5 Conclusions

This paper provides new complexity results for reasoning over Extended Entity-Relationship (EER) models with different modelling constructors. Starting from the ExpTime result [3] for reasoning over the fully-fledged EER language, we prove that the same complexity holds even if the Boolean constructors (disjointness and covering) on relationships are dropped. This result shows that isa between relationships (with the Booleans on entities) is powerful enough to capture ExpTime-hard problems. To illustrate that the presence of relationship hierarchies is a major source of complexity in reasoning we show that avoiding them makes reasoning in ER_{bool} an NP-complete problem. Another source of complexity is the covering constraint. Indeed, without relationship hierarchies and covering constraints reasoning problem for ER_{ref} is NLogSpace-complete.

The paper also provides a tight correspondence between conceptual modelling languages and the *DL-Lite* family of description logics and shows the usefulness of *DL-Lite* in representing and reasoning over conceptual models and ontologies.

References

1. Batini, C., Ceri, S., Navathe, S.B.: Conceptual Database Design, an Entity-Relationship Approach. Benjamin and Cummings Publ. Co. (1992)
2. ElMasri, R.A., Navathe, S.B.: Fundamentals of Database Systems. 5th edn. Addison Wesley Publ. Co. (2007)
3. Berardi, D., Calvanese, D., De Giacomo, G.: Reasoning on UML class diagrams. Artificial Intelligence **168**(1–2) (2005) 70–118
4. Calvanese, D., Lenzerini, M., Nardi, D.: Unifying class-based representation formalisms. J. of Artificial Intelligence Research **11** (1999) 199–240
5. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Data complexity of query answering in description logics. In: KR'06. (2006) 260–270
6. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: *DL-Lite* in the light of first-order logic. In: AAAI'07. (2007)
7. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: OWL Web Ontology Language reference. W3C Recommendation (February 2004) Available at `http://www.w3.org/TR/owl-ref/`.
8. Di Battista, G., Lenzerini, M.: Deductive entity-relationship modeling. IEEE Trans. on Knowledge and Data Engineering **5**(3) (1993) 439–450
9. Lenzerini, M., Nobili, P.: On the satisfiability of dependency constraints in entity-relationship schemata. Information Systems **15**(4) (1990) 453–461
10. Calvanese, D., Lenzerini, M.: On the interaction between ISA and cardinality constraints. In: ICDE'94. (1994) 204–213
11. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: DL-Lite: Tractable description logics for ontologies. In: AAAI'05. (2005) 602–607
12. Kozen, D.: Theory of Computation. Springer (2006)