Extending $\mathcal{DLR}$ with Labelled Tuples, Projections, Functional Dependencies and Objectification

Alessandro Artale and Enrico Franconi

KRDB Research Centre, Free University of Bozen-Bolzano, Italy
{artale,franconi}@inf.unibz.it

Abstract. We introduce an extension of the n-ary description logic $\mathcal{DLR}$ to deal with attribute-labelled tuples (generalising the positional notation), with arbitrary projections of relations (inclusion dependencies), generic functional dependencies and with global and local objectification (reifying relations or their projections). We show how a simple syntactic condition on the appearance of projections and functional dependencies in a knowledge base makes the language decidable without increasing the computational complexity of the basic $\mathcal{DLR}$ language.

1 Introduction

We introduce in this paper the language $\mathcal{DLR}^+$ which extends the n-ary description logics $\mathcal{DLR}$ [Calvanese *et al.*, 1998; Baader *et al.*, 2003] and $\mathcal{DLR}_{ifd}$ [Calvanese *et al.*, 2001] as follows:

- the semantics is based on attribute-labelled tuples: an element of a tuple is identified by an attribute and not by its position in the tuple, e.g., the relation `Person` has attributes `firstname`, `lastname`, `age`, `height` with instance: $\langle$ `firstname: Enrico, lastname: Franconi, age: 53, height: 1.90`$\rangle$;
- renaming of attributes is possible, e.g., to recover the positional semantics: `firstname,lastname,age,height` $\twoheadrightarrow$ `1,2,3,4`;
- it can express projections of relations, and therefore inclusion dependencies, e.g., $\exists$[`firstname,lastname`]`Student` $\sqsubseteq$ $\exists$[`firstname,lastname`]`Person`;
- it can express multiple-attribute cardinalities, and therefore functional dependencies and multiple-attribute keys, e.g., the functional dependency from `firstname, lastname` to `age` in `Person` can be written as:
  $\exists$[`firstname,lastname`]`Person` $\sqsubseteq$
  $\quad \exists^{\leqslant 1}$[`firstname,lastname`]($\exists$[`firstname,lastname,age`]`Person`);
- it can express global and local objectification (also known as reification): a tuple may be identified by a unique global identifier, or by an identifier which is unique only within the interpretation of a relation, e.g., to identify the name of a person we can write `Name` $\sqsubseteq \bigodot \exists$[`firstname,lastname`]`Person`.

We show how a simple syntactic condition on the appearance of projections in the knowledge base makes the language decidable without increasing the computational

$$
\begin{array}{rcl}
C & \rightarrow & \top \mid \bot \mid CN \mid \neg C \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \exists^{\leqslant q}[U_i]R \mid \circledcirc R \mid \odot RN \\
R & \rightarrow & RN \mid R_1 \backslash R_2 \mid R_1 \sqcap R_2 \mid R_1 \sqcup R_2 \mid \sigma_{U_i:C} R \mid \exists^{\leqslant q}[U_1, \ldots, U_k]R \\
\varphi & \rightarrow & C_1 \sqsubseteq C_2 \mid R_1 \sqsubseteq R_2 \\
\vartheta & \rightarrow & U_1 \leftrightarrow U_2
\end{array}
$$

Fig. 1. Syntax of $\mathcal{DLR}^+$.

$$
\begin{array}{rcll}
\tau(R_1 \backslash R_2) & = & \tau(R_1) & \text{if } \tau(R_1) = \tau(R_2) \\
\tau(R_1 \sqcap R_2) & = & \tau(R_1) & \text{if } \tau(R_1) = \tau(R_2) \\
\tau(R_1 \sqcup R_2) & = & \tau(R_1) & \text{if } \tau(R_1) = \tau(R_2) \\
\tau(\sigma_{U_i:C} R) & = & \tau(R) & \text{if } U_i \in \tau(R) \\
\tau(\exists^{\leqslant q}[U_1, \ldots, U_k]R) & = & \{U_1, \ldots, U_k\} & \text{if } \{U_1, \ldots, U_k\} \subset \tau(R) \\
\tau(R) & = & \varnothing & \text{otherwise}
\end{array}
$$

Fig. 2. The signature of $\mathcal{DLR}^+$ relations.

complexity of the basic $\mathcal{DLR}$ language. We call $\mathcal{DLR}^{\pm}$ this fragment of $\mathcal{DLR}^+$. $\mathcal{DLR}^{\pm}$ is able to correctly express the UML fragment as introduced in [Berardi *et al.*, 2005; Artale *et al.*, 2007] and the ORM fragment as introduced in [Franconi and Mosca, 2013].

2 Syntax of the Description Logic $\mathcal{DLR}^+$

We first define the syntax of the language $\mathcal{DLR}^+$. A *signature* in $\mathcal{DLR}^+$ is a triple $\mathcal{L} = (\mathcal{C}, \mathcal{R}, \mathcal{U}, \tau)$ consisting of a finite set $\mathcal{C}$ of *concept* names (denoted by CN), a finite set $\mathcal{R}$ of *relation* names (denoted by RN) disjoint from $\mathcal{C}$, and a finite set $\mathcal{U}$ of *attributes* (denoted by U), and a *relation signature* function τ associating a set of attributes to each relation name, $\tau(RN) = \{U_1, \ldots, U_n\} \subseteq \mathcal{U}$ with $n \geqslant 2$.

The syntax of concepts C, relations R, formulas φ, and attribute renaming axioms ϑ is defined in Figure 1, where q is a positive integer and $2 \leqslant k < \text{ARITY}(R)$. We extend the signature function τ to arbitrary relations as specified in Figure 2. We define the ARITY of a relation R as the number of the attributes in its signature, namely $|\tau(R)|$.

A $\mathcal{DLR}^+$ *TBox* $\mathcal{T}$ is a finite set of formulas, i.e., *concept inclusion* axioms of the form $C_1 \sqsubseteq C_2$ and *relation inclusion* axioms of the form $R_1 \sqsubseteq R_2$.
A renaming schema induces an equivalence relation $(\leftrightarrow, \mathcal{U})$ over the attributes $\mathcal{U}$, providing a partition of $\mathcal{U}$ into equivalence classes each one representing the alternative ways to name attributes. We write $[U]_{\Re}$ to denote the equivalence class of the attribute U w.r.t. the equivalence relation $(\leftrightarrow, \mathcal{U})$. We allow only *well founded* renaming schemas, namely schemas such that each equivalence class $[U]_{\Re}$ in the induced equivalence relation never contains two attributes from the same relation signature. In the following we use the shortcut $U_1 \ldots U_n \leftrightarrow U'_1 \ldots U'_n$ to group many renaming axioms, with the obvious meaning that $U_i \leftrightarrow U'_i$, for all $i = 1, \ldots, n$.

A $\mathcal{DLR}^+$ knowledge base $\mathcal{KB} = (\mathcal{T}, \Re)$ is composed by a TBox $\mathcal{T}$ and a renaming schema $\Re$.

The renaming schema reconciles the attribute and the positional perspectives on relations (see also the similar perspectives in relational databases [Abiteboul *et al.*, 1995]). They are crucial when expressing both inclusion axioms and operators ($\sqcap$, $\sqcup$, $\setminus$) between relations, which make sense only over *union compatible* relations. Two relations R_1, R_2 are union compatible if their signatures are equal up to the attribute renaming induced by the renaming schema $\Re$, namely, $\tau(R_1) = \{U_1, \ldots, U_n\}$ and $\tau(R_2) = \{V_1, \ldots, V_n\}$ have the same arity n and $[U_i]_\Re = [V_i]_\Re$ for each $1 \leqslant i \leqslant n$. Notice that, thanks to the renaming schema, relations can use just local attribute names that can then be renamed when composing relations. Also note that it is obviously possible for the same attribute to appear in the signature of different relations.

To show the expressive power of the language, let us consider the following example with tree relation names R_1, R_2 and R_3 with the following signature:

$$\begin{aligned}\tau(R_1) &= \{U_1, U_2, U_3, U_4, U_5\}\\ \tau(R_2) &= \{V_1, V_2, V_3, V_4, V_5\}\\ \tau(R_3) &= \{W_1, W_2, W_3, W_4\}\end{aligned}$$

To state that $\{U_1, U_2\}$ is the *multi-attribute key* of R_1 we add the axiom:

$$\exists[U_1, U_2]R_1 \sqsubseteq \exists^{\leqslant 1}[U_1, U_2]R_1$$

where $\exists[U_1, \ldots, U_k]R$ stands for $\exists^{\geqslant 1}[U_1, \ldots, U_k]R$. To express that there is a *functional dependency* from the attributes $\{V_3, V_4\}$ to the attribute $\{V_5\}$ of R_2 we add the axiom:

$$\exists[V_3, V_4]R_2 \sqsubseteq \exists^{\leqslant 1}[V_3, V_4](\exists[V_3, V_4, V_5]R_2) \tag{1}$$

The following axioms express that R_2 is a sub-relation of R_1 and that a projection of R_3 is a sub-relation of a projection of R_1, together with the corresponding axioms for the renaming schema to explicitly specify the correspondences between the attributes of the two inclusion dependencies:

$$\begin{aligned}R_2 &\sqsubseteq R_1\\ \exists[W_1, W_2, W_3]R_3 &\sqsubseteq \exists[U_3, U_4, U_5]R_1\\ V_1V_2V_3V_4V_5 &\twoheadrightarrow U_1U_2U_3U_4U_5\\ W_1W_2W_3 &\twoheadrightarrow U_3U_4U_5\end{aligned}$$

3 Semantics

The semantics makes use of the notion of *labelled tuples* over a domain set Δ: a *$\mathcal{U}$-labelled tuple over Δ* is a function $t\colon \mathcal{U} \to \Delta$. For $U \in \mathcal{U}$, we write $t[U]$ to refer to the domain element $d \in \Delta$ labelled by U, if the function t is defined for U – that is, if the attribute U is a label of the tuple t. Given $d_1, \ldots, d_n \in \Delta$, the expression

$$
\begin{aligned}
\top^{\mathcal{I}} &= \Delta \\
\bot^{\mathcal{I}} &= \varnothing \\
(\neg C)^{\mathcal{I}} &= \top^{\mathcal{I}} \backslash C^{\mathcal{I}} \\
(C_1 \sqcap C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}} \\
(C_1 \sqcup C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}} \\
(\exists^{\lessgtr q}[U_i]R)^{\mathcal{I}} &= \{d \in \Delta \mid \ \big|\{t \in R^{\mathcal{I}} \mid t[\rho(U_i)] = d\}\big| \lessgtr q\} \\
(\circledcirc\, R)^{\mathcal{I}} &= \{d \in \Delta \mid d = \imath(t) \wedge t \in R^{\mathcal{I}}\} \\
(\odot\, RN)^{\mathcal{I}} &= \{d \in \Delta \mid d = \ell_{RN}(t) \wedge t \in RN^{\mathcal{I}}\}
\end{aligned}
$$

$$
\begin{aligned}
(R_1 \backslash R_2)^{\mathcal{I}} &= R_1^{\mathcal{I}} \backslash R_2^{\mathcal{I}} \\
(R_1 \sqcap R_2)^{\mathcal{I}} &= R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}} \\
(R_1 \sqcup R_2)^{\mathcal{I}} &= \{t \in R_1^{\mathcal{I}} \cup R_2^{\mathcal{I}} \mid \rho(\tau(R_1)) = \rho(\tau(R_2))\} \\
(\sigma_{U_i:C} R)^{\mathcal{I}} &= \{t \in R^{\mathcal{I}} \mid t[\rho(U_i)] \in C^{\mathcal{I}}\} \\
(\exists^{\lessgtr q}[U_1, \ldots, U_k]R)^{\mathcal{I}} &= \{\langle \rho(U_1) : d_1, \ldots, \rho(U_k) : d_k \rangle \in T_\Delta(\{\rho(U_1), \ldots, \rho(U_k)\}) \mid \\
&\qquad \big|\{t \in R^{\mathcal{I}} \mid t[\rho(U_1)] = d_1, \ldots, t[\rho(U_k)] = d_k\}\big| \lessgtr q\}
\end{aligned}
$$

Fig. 3. Semantics of $\mathcal{DLR}^+$ expressions.

$\langle U_1 \colon d_1, \ldots, U_n \colon d_n \rangle$ stands for the $\mathcal{U}$-labelled tuple t over Δ (tuple, for short) such that $t[U_i] = d_i$, for $1 \leqslant 1 \leqslant n$. We write $t[U_1, \ldots, U_k]$ to denote the *projection* of the tuple t over the attributes $U_1, \ldots, U_k$, namely the function t restricted to be undefined for the labels not in $U_1, \ldots, U_k$. The set of all $\mathcal{U}$-labelled tuples over Δ is denoted by $T_\Delta(\mathcal{U})$.

A $\mathcal{DLR}^+$ *interpretation*, $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}}, \rho, \imath, \ell_{RN_1}, \ell_{RN_2}, \ldots)$, consists of a nonempty *domain* Δ, an *interpretation function* $\cdot^{\mathcal{I}}$, a *renaming function* ρ, a *global objectification function* $\imath$, and a family of *local objectification functions* ℓ_{RN_i}, one for each named relation $RN_i \in \mathcal{R}$.

The renaming function ρ for attributes is a total function $\rho : \mathcal{U} \to \mathcal{U}$ representing a canonical renaming for all attributes. We consider, as a shortcut, the notation $\rho(\{U_1, \ldots, U_k\}) = \{\rho(U_1), \ldots, \rho(U_k)\}$.

The global objectification function is an injective function, $\imath : T_\Delta(\mathcal{U}) \to \Delta$, associating a *unique* global identifier to each possible tuple.

The local objectification functions, $\ell_{RN_i} : T_\Delta(\mathcal{U}) \to \Delta$, are distinct for each relation name in the signature, and as the global objectification function they are injective: they associate an identifier – which is unique only within the interpretation of a relation name – to each possible tuple.

The interpretation function $\cdot^{\mathcal{I}}$ assigns a set of domain elements to each concept name, $CN^{\mathcal{I}} \subseteq \Delta$, and a set of $\mathcal{U}$-labelled tuples over Δ to each relation name conforming with its signature and the renaming function:

$$RN^{\mathcal{I}} \subseteq T_\Delta(\{\rho(U) \mid U \in \tau(RN)\}).$$

The interpretation function $\cdot^{\mathcal{I}}$ is unambiguously extended over concept and relation expressions as specified in the inductive definition of Fig. 3.

An interpretation $\mathcal{I}$ satisfies a concept inclusion axiom $C_1 \sqsubseteq C_2$ if $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$, it satisfies a relation inclusion axiom $R_1 \sqsubseteq R_2$ if $R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$, and it satisfies a renaming schema $\Re$ if the renaming function ρ renames the attributes in a consistent way with respect to $\Re$, namely if

$$\forall U . \rho(U) \in [U]_{\Re} \wedge \forall V \in [U]_{\Re} . \rho(U) = \rho(V).$$

An interpretation is a *model* for a knowledge base $(\mathcal{T}, \Re)$ if it satisfies all the formulas in the TBox $\mathcal{T}$ and it satisfies the renaming schema $\Re$. We define *KB satisfiability* as the problem of deciding the existence of a model of a given knowledge base, *concept satisfiability* (resp. *relation satisfiability*) as the problem of deciding whether there is a model of the knowledge base that assigns a non-empty extension to a given concept (resp. relation), and *entailment* as the problem to check whether a given knowledge base logically implies a formula, that is, whenever all the models of the knowledge base are also models of the formula.
For example, from the knowledge base $\mathcal{KB}$ introduced in the previous Section the following logical implication holds:

$$\mathcal{KB} \models \exists[V_1, V_2]R_2 \sqsubseteq \exists^{\leqslant 1}[V_1, V_2]R_2$$

i.e., the attributes V_1, V_2 are a key for the relation R_2.

Proposition 1. *The problems of KB satisfiability, concept and relation satisfiability, and entailment are mutually reducible in* $\mathcal{DLR}^+$.

$\mathcal{DLR}^+$ can express complex inclusion and functional dependencies, for which it is well known that reasoning is undecidable [Mitchell, 1983; Chandra and Vardi, 1985]. $\mathcal{DLR}^+$ also includes the $\mathcal{DLR}$ extension $\mathcal{DLR}_{ifd}$ together with unary functional dependencies [Calvanese *et al.*, 2001], which also has been proved to be undecidable.

4 The $\mathcal{DLR}^{\pm}$ fragment of $\mathcal{DLR}^+$

Given a $\mathcal{DLR}^+$ knowledge base $(\mathcal{T}, \Re)$, we define the *projection signature* as the set $\mathscr{T}$ including the signatures $\tau(R\!N)$ of the relations $R\!N \in \mathcal{R}$, the singletons associated with each attribute name $U \in \mathcal{U}$, and the relation signatures as they appear explicitly in projection constructs in the relation inclusion axioms of the knowledge base, together with their implicit occurrences due to the renaming schema:

1. $\tau(R\!N) \in \mathscr{T}$ if $R\!N \in \mathcal{R}$;
2. $\{U\} \in \mathscr{T}$ if $U \in \mathcal{U}$;
3. $\{U_1, \ldots, U_k\} \in \mathscr{T}$ if $\exists^{\leqslant q}[V_1, \ldots, V_k]R \in \mathcal{T}$ and $\{U_i, V_i\} \subseteq [U_i]_{\Re}$ for $1 \leqslant i \leqslant k$.

We call *projection signature graph* the directed acyclic graph $(\supset, \mathscr{T})$ with the attribute singletons $\{U\}$ being the sinks. The $\mathcal{DLR}^{\pm}$ fragment of $\mathcal{DLR}^+$ allows only for knowledge bases with a projection signature graph being a *multitree*, namely the set of nodes reachable from any node of the projection signature graph should form a tree.

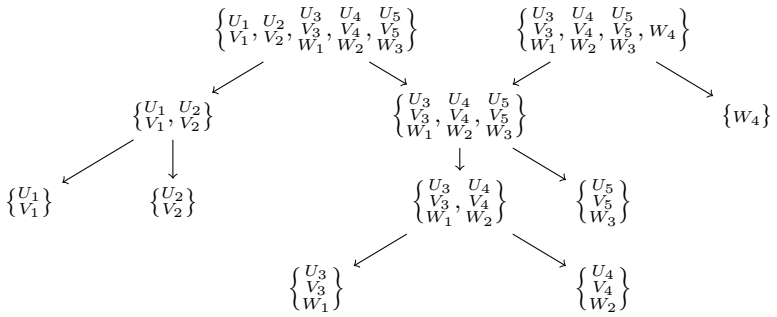

Fig. 4. The projection signature graph of the example.

Given a relation name RN, the subgraph of the projection signature graph dominated by RN is a tree where the leaves are all the attributes in $\tau(RN)$ and the root is $\tau(RN)$. We call $\mathscr{T}_{\{U_1,\ldots,U_k\}}$ the tree formed by the nodes in the projection signature graph dominated by the set of attributes $\{U_1,\ldots,U_k\}$. Given two relation signatures (i.e., two sets of attributes) $\tau_1, \tau_2 \subseteq \mathcal{U}$, by $\text{PATH}_{\mathscr{T}}(\tau_1,\tau_2)$ we denote the path in $(\supset, \mathscr{T})$ between τ_1 and τ_2, if it exists. Note that $\text{PATH}_{\mathscr{T}}(\tau_1,\tau_2) = \varnothing$ both when a path does not exist and when $\tau_1 \subseteq \tau_2$, and $\text{PATH}_{\mathscr{T}}$ is functional in $\mathcal{DLR}^{\pm}$ due to the multitree restriction on projection signatures. The notation $\text{CHILD}_{\mathscr{T}}(\tau_1,\tau_2)$ means that τ_2 is a child of τ_1 in $(\supset, \mathscr{T})$.

In addition to the above multitree condition, the $\mathcal{DLR}^{\pm}$ fragment of $\mathcal{DLR}^{+}$ allows for knowledge bases with projection constructs $\exists^{\leqslant q}[U_1,\ldots,U_k]R$ (resp. $\exists^{\leqslant q}[U]R$) with a cardinality $q > 1$ only if the length of the path $\text{PATH}_{\mathscr{T}}(\{U_1,\ldots,U_k\},\tau(R))$ (resp. $\text{PATH}_{\mathscr{T}}(\{U\},\tau(R))$) is 1. This allows to map cardinalities in $\mathcal{DLR}^{\pm}$ into cardinalities in $\mathcal{ALCQI}$.

Figure 4 shows that the projection signature graph of the knowledge base introduced in Section 2 is indeed a multitree. Note that in the figure we have collapsed equivalent attributes in a unique equivalence class, according to the renaming schema.

$\mathcal{DLR}^{\pm}$ restricts $\mathcal{DLR}^{+}$ only in the way multiple projections of relations appear in the knowledge base. It is easy to see that $\mathcal{DLR}$ is included in $\mathcal{DLR}^{\pm}$, since the projection signature graph of any $\mathcal{DLR}$ knowledge base has maximum depth equal to 1. $\mathcal{DLR}_{\mathit{ifd}}$ [Calvanese *et al.*, 2001] together with (unary) functional dependencies is also included in $\mathcal{DLR}^{\pm}$, with the proviso that projections of relations in the knowledge base form a multitree projection signature graph. Since (unary) functional dependencies are expressed via the inclusions of projections of relations (see, e.g., the functional dependency (1) in the previous example), by constraining the projection signature graph to be a multitree, the possibility to build combinations of functional dependencies as the ones in [Calvanese *et al.*, 2001] leading to undecidability is ruled out. Also note that $\mathcal{DLR}^{\pm}$ is able to correctly express the UML fragment as introduced in [Berardi *et al.*, 2005; Artale *et al.*, 2007] and the ORM fragment as introduced in [Franconi and Mosca, 2013].

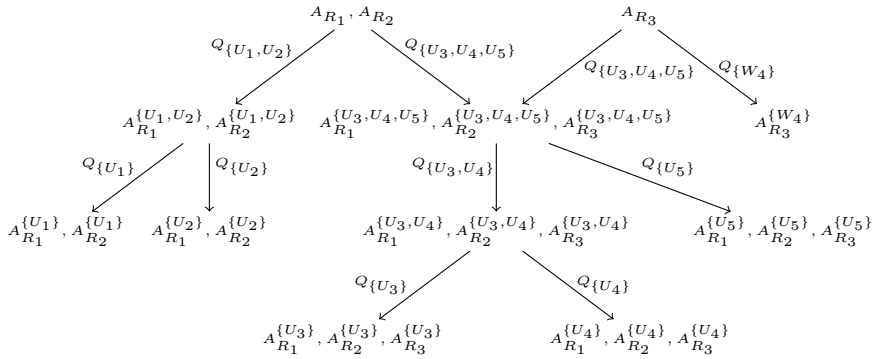

Fig. 5. The $\mathcal{ALCQI}$ signature generated by the example.

5 Mapping $\mathcal{DLR}^{\pm}$ to $\mathcal{ALCQI}$

We show that reasoning in $\mathcal{DLR}^{\pm}$ is EXPTIME-complete by providing a mapping from $\mathcal{DLR}^{\pm}$ knowledge bases to $\mathcal{ALCQI}$ knowledge bases; the reverse mapping from $\mathcal{ALCQI}$ knowledge bases to $\mathcal{DLR}$ knowledge bases is well known. The proof is based on the fact that reasoning with $\mathcal{ALCQI}$ knowledge bases is EXPTIME-complete [Baader *et al.*, 2003]. We adapt and extend the mapping presented for $\mathcal{DLR}$ in [Calvanese *et al.*, 1998].

In the following we use the shortcut $(S_1 \circ \ldots \circ S_n)^-$ for $S_n^- \circ \ldots \circ S_1^-$, the shortcut $\exists^{\leqslant 1} S_1 \circ \ldots \circ S_n.C$ for $\exists^{\leqslant 1} S_1.\ldots.\exists^{\leqslant 1} S_n.C$ and the shortcut $\forall S_1 \circ \ldots \circ S_n.C$ for $\forall S_1.\ldots.\forall S_n.C$. Note that these shortcuts for the role chain constructor "$\circ$" are not correct in general, but they are correct in the context of the specific $\mathcal{ALCQI}$ knowledge bases used in this paper.

Let $\mathcal{KB} = (\mathcal{T}, \Re)$ be a $\mathcal{DLR}^{\pm}$ knowledge base. We first rewrite the knowledge base as follows: for each equivalence class $[U]_{\Re}$ a single *canonical* representative of the class is chosen, and the $\mathcal{KB}$ is consistently rewritten by substituting each attribute with its canonical representative. After this rewriting, the renaming schema does not play any role in the mapping.

The mapping function $\cdot^{\dagger}$ maps each concept name *CN* in the $\mathcal{DLR}^{\pm}$ knowledge base to an $\mathcal{ALCQI}$ concept name *CN*, each relation name *RN* in the $\mathcal{DLR}^{\pm}$ knowledge base to an $\mathcal{ALCQI}$ concept name A_{RN} (its global reification), and each attribute name U in the $\mathcal{DLR}^{\pm}$ knowledge base to an $\mathcal{ALCQI}$ role name, as detailed below.
For each relation name *RN* the mapping introduces a concept name A^{l}_{RN} and a role name Q_{RN} (to capture the local reification), and a concept name $A^{\tau_i}_{RN}$ for each projected signature τ_i in the projection signature graph dominated by $\tau(RN)$, $\tau_i \in \mathscr{T}_{\tau(RN)}$ (to capture global reifications of the projections of *RN*). Note that $A^{\tau(RN)}_{RN}$ coincides with A_{RN}. Furthermore, the mapping introduces a role name Q_{τ_i} for each projected signature τ_i in the projection signature, $\tau_i \in \mathscr{T}$, such that there exists $\tau_j \in \mathscr{T}$ with $\text{CHILD}_{\mathscr{T}}(\tau_j, \tau_i)$, i.e., we exclude the case where τ_i is one of the roots of the multitree induced by the projection signature.

$$
\begin{aligned}
(\neg C)^\dagger &= \neg C^\dagger \\
(C_1 \sqcap C_2)^\dagger &= C_1^\dagger \sqcap C_2^\dagger \\
(C_1 \sqcup C_2)^\dagger &= C_1^\dagger \sqcup C_2^\dagger \\
(\exists^{\trianglerighteq q}[U_i]R)^\dagger &= \exists^{\trianglerighteq q} \left(\text{PATH}_{\mathscr{T}}(\tau(R), \{U_i\})^\dagger\right)^- . R^\dagger \\
(\circledcirc R)^\dagger &= R^\dagger \\
(\odot RN)^\dagger &= A^l_{RN}
\end{aligned}
$$

$$
\begin{aligned}
(R_1 \backslash R_2)^\dagger &= R_1^\dagger \sqcap \neg R_2^\dagger \\
(R_1 \sqcap R_2)^\dagger &= R_1^\dagger \sqcap R_2^\dagger \\
(R_1 \sqcup R_2)^\dagger &= R_1^\dagger \sqcup R_2^\dagger \\
(\sigma_{U_i:C} R)^\dagger &= R^\dagger \sqcap \forall \text{PATH}_{\mathscr{T}}(\tau(R), \{U_i\})^\dagger . C^\dagger \\
(\exists^{\trianglerighteq q}[U_1, \ldots, U_k]R)^\dagger &= \exists^{\trianglerighteq q} \left(\text{PATH}_{\mathscr{T}}(\tau(R), \{U_1, \ldots, U_k\})^\dagger\right)^- . R^\dagger
\end{aligned}
$$

Fig. 6. The mapping for concept and relation expressions.

The mapping $\cdot^\dagger$ applies also to a path. Let $\tau, \tau' \in \mathscr{T}$ be two generic sets of attributes such that the function $\text{PATH}_{\mathscr{T}}(\tau, \tau') = \tau, \tau_1, \ldots, \tau_n, \tau'$, then, a path is mapped as follows:

$$\text{PATH}_{\mathscr{T}}(\tau, \tau')^\dagger = Q_{\tau_1} \circ \ldots \circ Q_{\tau_n} \circ Q_{\tau'}.$$

Intuitively, the mapping reifies each node in the projection signature graph: the target $\mathcal{ALCQI}$ signature of the example of the previous section is partially presented in Fig. 5, together with the projection signature graph. Each node is labelled with the corresponding (global) reification concept ($A^{\tau_j}_{R_i}$), for each relation name R_i and each projected signature τ_j in the projection signature graph dominated by $\tau(R_i)$, while the edges are labelled by the roles (Q_{τ_i}) needed for the reification.

The mapping $\cdot^\dagger$ is extended to concept and relation expressions as in Figure 6, with the proviso that whenever $\text{PATH}_{\mathscr{T}}(\tau_1, \tau_2)$ returns an empty path then the translation for the corresponding expression becomes the bottom concept. Note that in $\mathcal{DLR}^\pm$ the cardinalities on a path are restricted to the case $q = 1$ whenever a path is of length greater than 1, so we still remain within the $\mathcal{ALCQI}$ description logic when the mapping applies to cardinalities. So, if we need to express a cardinality constraint $\exists^{\trianglerighteq q}[U_i]R,]$ with $q > 1$, then U_i should not be mentioned in any other projection of the relation R in such a way that $|\text{PATH}_{\mathscr{T}}(\tau(R), \{U_i\})| = 1$.

In order to explain the need for the path function in the mapping, notice that a relation is reified according to the decomposition dictated by projection signature graph it dominates. Thus, to access an attribute U_j of a relation R_i it is necessary to follow the path through the projections that use that attribute. This path is a role chain from the signature of the relation (the root) to the attribute as returned by the $\text{PATH}_{\mathscr{T}}(\tau(R_i), U_i)$ function. For example, considering Fig. 5, in order to access the attribute U_4 of the relation R_3 in the expression $(\sigma_{U_4:C} R_3)$, the path $\text{PATH}_{\mathscr{T}}(\tau(R_3), \{U_4\})^\dagger$ is equal to the role chain $Q_{\{U_3,U_4,U_5\}} \circ Q_{\{U_3,U_4\}} \circ Q_{\{U_4\}}$, so that $(\sigma_{U_4:C} R_3)^\dagger = A_{R_3} \sqcap \forall Q_{\{U_3,U_4,U_5\}} \circ Q_{\{U_3,U_4\}} \circ Q_{\{U_4\}} . C$.

Similar considerations can be done when mapping cardinalities over relation projections.

The mapping $\gamma(\mathcal{KB})$ of a $\mathcal{DLR}^{\pm}$ knowledge base $\mathcal{KB}$ with a signature $(\mathcal{C}, \mathcal{R}, \mathcal{U}, \tau)$ is defined as the following $\mathcal{ALCQI}$ TBox:

$$\gamma(\mathcal{KB}) \quad = \quad \gamma_{dsj} \cup \bigcup_{RN \in \mathcal{R}} \gamma_{rel}(RN) \cup \bigcup_{RN \in \mathcal{R}} \gamma_{lobj}(RN) \cup \bigcup_{C_1 \sqsubseteq C_2 \in \mathcal{KB}} C_1^{\dagger} \sqsubseteq C_2^{\dagger} \cup \bigcup_{R_1 \sqsubseteq R_2 \in \mathcal{KB}} R_1^{\dagger} \sqsubseteq R_2^{\dagger}$$

where

$$\gamma_{dsj} = \left\{ A_{RN_1}^{\tau_i} \sqsubseteq \neg A_{RN_2}^{\tau_j} \mid RN_1, RN_2 \in \mathcal{R}, \tau_i, \tau_j \in \mathscr{T}, |\tau_i| \geqslant 2, |\tau_j| \geqslant 2, \tau_i \neq \tau_j \right\}$$

$$\gamma_{rel}(RN) = \bigcup_{\tau_i \in \mathscr{T}_{\tau(RN)}} \bigcup_{\text{CHILD}_{\mathscr{T}}(\tau_i, \tau_j)} \left\{ A_{RN}^{\tau_i} \sqsubseteq \exists Q_{\tau_j}.A_{RN}^{\tau_j},\ \exists^{\geqslant 2} Q_{\tau_j}.\top \sqsubseteq \bot \right\}$$

$$\begin{aligned} \gamma_{lobj}(RN) = \{ & A_{RN} \sqsubseteq \exists Q_{RN}.A_{RN}^{l},\ \exists^{\geqslant 2} Q_{RN}.\top \sqsubseteq \bot, \\ & A_{RN}^{l} \sqsubseteq \exists Q_{RN}^{-}.A_{RN},\ \exists^{\geqslant 2} Q_{RN}^{-}.\top \sqsubseteq \bot \}. \end{aligned}$$

Intuitively, γ_{dsj} ensures that relations with different signatures are disjoint, thus, e.g., enforcing the union compatibility. The axioms in γ_{rel} introduce classical reification axioms for each relation and its relevant projections. The axioms in γ_{lobj} make sure that each local objectification differs form the global one.

Clearly, the size of $\gamma(\mathcal{KB})$ is polynomial in the size of $\mathcal{KB}$ (under the same coding of the numerical parameters), and thus we are able to state the main result of this paper (see [Artale and Franconi, 2016] for a complete set of proofs of the Theorem).

Theorem 2. *A $\mathcal{DLR}^{\pm}$ knowledge base $\mathcal{KB}$ is satisfiable iff the $\mathcal{ALCQI}$ knowledge base $\gamma(\mathcal{KB})$ is satisfiable.*

As a direct consequence of the above theorem and the fact that $\mathcal{DLR}$ is a sublanguage of $\mathcal{DLR}^{\pm}$, we have that

Corollary 3. *Reasoning in $\mathcal{DLR}^{\pm}$ is an* EXPTIME*-complete problem.*

6 Acknowledgements

We thank Alessandro Mosca for working with us on all the preliminary work necessary to understand how to get these technical results.

References

[Abiteboul *et al.*, 1995] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases.* Addison-Wesley, 1995.

[Artale and Franconi, 2016] Alessandro Artale and Enrico Franconi. Extending DLR with labelled tuples, projections, functional dependencies and objectification (full version). *CoRR*, abs/1604.00799, April 2016.

[Artale *et al.*, 2007] A. Artale, D. Calvanese, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. Reasoning over extended ER models. In *Proc. of the 26th Int. Conf. on Conceptual Modeling (ER'07)*, volume 4801 of *Lecture Notes in Computer Science*, pages 277–292. Springer, 2007.

[Baader *et al.*, 2003] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

[Berardi *et al.*, 2005] D. Berardi, D. Calvanese, and G. De Giacomo. Reasoning on UML class diagrams. *Artificial Intelligence*, 168(1–2):70–118, 2005.

[Calvanese *et al.*, 1998] D. Calvanese, G. De Giacomo, and M. Lenzerini. On the decidability of query containment under constraints. In *Proc. of the 17th ACM Sym. on Principles of Database Systems (PODS'98)*, pages 149–158, 1998.

[Calvanese *et al.*, 2001] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. Identification constraints and functional dependencies in description logics. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI-01*, pages 155–160. Morgan Kaufmann, 2001.

[Chandra and Vardi, 1985] Ashok K. Chandra and Moshe Y. Vardi. The implication problem for functional and inclusion dependencies is undecidable. *SIAM Journal on Compututing*, 14(3):671–677, 1985.

[Franconi and Mosca, 2013] Enrico Franconi and Alessandro Mosca. Towards a core ORM2 language (research note). In *OTM Workshops*, volume 8186 of *Lecture Notes in Computer Science*, pages 448–456. Springer, 2013.

[Mitchell, 1983] John C. Mitchell. The implication problem for functional and inclusion dependencies. *Information and Control*, 56(3):154–173, 1983.