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

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] 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:
  \[
  \langle \text{firstname: Enrico, lastname: Franconi, age: 53, height: 1.90} \rangle
  \]
- renaming of attributes is possible, e.g., to recover the positional semantics:
  \[
  \text{firstname, lastname, age, height} \leftrightarrow 1, 2, 3, 4;
  \]
- it can express projections of relations, and therefore inclusion dependencies, e.g.,
  \[
  \exists\text{[firstname, lastname]} \text{Student} \subseteq \exists\text{[firstname, lastname]} \text{Person}.
  \]
- 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:
  \[
  \exists\text{[firstname, lastname]} \text{Person} \subseteq \\
  \exists^1\text{[firstname, lastname]} \exists\text{[firstname, lastname, age]} \text{Person};
  \]
- 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 $\subseteq \bigcirc \exists\text{[firstname, lastname]} \text{Person}.$
We show how a simple syntactic condition on the appearance of projections in the knowledge base makes the language decidable without increasing the computational 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 $\tau$ associating a set of attributes to each relation name. $\tau(RN) = \{U_1, \ldots, U_n\} \subseteq \mathcal{U}$ with $n \geq 2$.

The syntax of concepts $C$, relations $R$, formulas $\varphi$, and attribute renaming axioms $\vartheta$ is defined in Figure 1 where $q$ is a positive integer and $2 \leq k < \text{arity}(R)$. We extend the signature function $\tau$ to arbitrary relations as specified in Figure 2. We define the $\text{arity}$ 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 $(\equiv, \mathcal{U})$ 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 $[U]_\mathcal{R}$ to denote the equivalence class of the attribute $U$ w.r.t. the equivalence relation $(\equiv, \mathcal{U})$. We allow only well founded renaming schemas, namely schemas such that each equivalence class $[U]_\mathcal{R}$ in the induced equivalence relation never contains two attributes from the same relation signature. In the
following we use the shortcut $U_1 \ldots U_n \mapsto U'_1 \ldots U'_n$ to group many renaming axioms, with the obvious meaning that $U_i \mapsto U'_i$, for all $i = 1, \ldots, n$.

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

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 ($\cap$, $\cup$, $\setminus$) 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 $\mathcal{R}$, namely, $\tau(R_1) = \{U_1, \ldots, U_n\}$ and $\tau(R_2) = \{V_1, \ldots, V_n\}$ have the same arity $n$ and $[U_i]_\mathcal{R} = [V_i]_\mathcal{R}$ for each $1 \leq i \leq n$.

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:

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

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

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

where $\exists[U_1, \ldots, U_k]R$ stands for $\exists^{\leq 1}[U_1, \ldots, U_k]R$. 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:

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

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:

$$
\begin{align*}
R_2 &\subseteq R_1 \\
\exists[W_1, W_2, W_3]R_3 &\subseteq \exists[U_3, U_4]R_1 \\
V_1V_2V_3V_4V_5 &\mapsto U_1U_2U_3U_4U_5 \\
W_1W_2W_3 &\mapsto U_3U_4U_5
\end{align*}
$$

3 Semantics

The semantics makes use of the notion of labelled tuples over a domain set $\Delta$: a $U$-labelled tuple over $\Delta$ is a function $t : U \rightarrow \Delta$. For $U \in \mathcal{U}$, we write $t[U]$ to refer
The interpretation function – to each possible tuple.

associate an identifier – which is unique only within the interpretation of a relation name

name in the signature, and as the global objectification function they are injective: they

CN

with its signature and the renaming function:

a

for the labels not in

that

The global objectification function is an injective function,

ť

ing a canonical renaming for all attributes. We consider, as a shortcut, the notation

T

∆

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

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

$\langle U_1 : d_1, \ldots, U_n : d_n \rangle$ stands for the $U$-labelled tuple $t$ over $\Delta$ (tuple, for short) such

that $t[U_i] = d_i$, for $1 \leq i \leq n$. 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 $U$-labelled tuples over $\Delta$ is denoted by

$T_\Delta(U)$.

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

The renaming function $\rho$ for attributes is a total function $\rho : U \rightarrow U$ 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, $\iota : T_\Delta(U) \rightarrow \Delta$, associating
a unique global identifier to each possible tuple.

The local objectification functions, $\ell_{RN_i} : T_\Delta(U) \rightarrow \Delta$, 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 $\mathcal{I}$ assigns a set of domain elements to each concept name,
$CN^\mathcal{I} \subseteq \Delta$, and a set of $U$-labelled tuples over $\Delta$ to each relation name conforming
with its signature and the renaming function:

$RN^\mathcal{I} \subseteq T_\Delta(\{\rho(U) \mid U \in \iota(RN)\})$. 
The interpretation function $^\mathcal{I}$ 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 $C_1^I \sqsubseteq C_2^I$, it satisfies a relation inclusion axiom $R_1 \sqsubseteq R_2$ if $R_1^I \subseteq R_2^I$, and it satisfies a renaming schema $^\mathcal{R}$ if the renaming function $\rho$ renames the attributes in a consistent way with respect to $^\mathcal{R}$, namely if

$$\forall U, \rho(U) \in [U]^R \land \forall V \in [U]^R, \rho(U) = \rho(V).$$

An interpretation is a model for a knowledge base $(^\mathcal{T}, ^\mathcal{R})$ if it satisfies all the formulas in the TBox $^\mathcal{T}$ and it satisfies the renaming schema $^\mathcal{R}$. 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 $KB$ introduced in the previous Section the following logical implication holds:

$$KB \models \exists [V_1, V_2]R_2 \sqsubseteq \exists^1 [V_1, V_2]R_2$$

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 $DLR^+$.  

**Proof.** We first show that we can reduce all the problems to concept satisfiability, where a concept $C$ is satisfiable iff $KB \not\models C \sqsubseteq \bot$.

- $KB$ is satisfiable iff $KB \not\models \top \sqsubseteq \bot$;
- $KB \models C_1 \sqsubseteq C_2$ iff $KB \models C_1 \land \neg C_2 \sqsubseteq \bot$;
- $KB \models R_1 \sqsubseteq R_2$ iff $KB \models \exists U (R_1 \cap \neg R_2) \sqsubseteq \bot$, for some $U \in \tau(R_1)$;
- $KB \not\models R \sqsubseteq \bot$ iff $KB \not\models \exists U [U]R \sqsubseteq \bot$, for some $U \in \tau(R)$.

Viceversa, we can show that concept satisfiability can be reduced to any other problem. First, note that concept satisfiability is already expressed as a logical implication problem. For the other cases, given a fresh new binary relation $P$, we have that

- $KB \not\models C \sqsubseteq \bot$ iff $KB \cup \{ \top \sqsubseteq \exists [U_1](P \cap \sigma_{U_2:C}P) \}$ is satisfiable;
- $KB \not\models C \sqsubseteq \bot$ iff $KB \not\models \sigma_{U_2:C}P \sqsubseteq \bot$.  

$DLR^+$ can express complex inclusion and functional dependencies, for which it is well known that reasoning is undecidable [Mitchell, 1983; Chandra and Vardi, 1985]. $DLR^+$ also includes the $DLR$ extension $DLR_{iif}$ 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 $(\mathcal{T}, \mathcal{R})$, we define the projection signature as the set $\mathcal{F}$ including the signatures $\tau(RN)$ of the relations $RN \in \mathcal{R}$, 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. $\tau(RN) \in \mathcal{F}$ if $RN \in \mathcal{R}$;
2. $\{U\} \in \mathcal{F}$ if $U \in \mathcal{U}$;
3. $\{U_1, \ldots, U_k\} \in \mathcal{F}$ if there exist $\exists q[V_1, \ldots, V_k]R \in \mathcal{T}$ and $\{U_i, V_i\} \subseteq [U_i]_\mathcal{R}$ for $1 \leq i \leq k$.

We call projection signature graph the directed acyclic graph $(\rightarrow, \mathcal{F})$ 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. 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 $\mathcal{F}(U_1, \ldots, U_k)$ 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 $\text{PATH}_\mathcal{F}(\tau_1, \tau_2)$ we denote the path in $(\rightarrow, \mathcal{F})$ between $\tau_1$ and $\tau_2$, if it exists. Note that $\text{PATH}_\mathcal{F}(\tau_1, \tau_2) = \emptyset$ both when a path does not exist and when $\tau_1 \subseteq \tau_2$, and $\text{PATH}_\mathcal{F}$ is functional in $\mathcal{DLR}^\pm$ due to the multitree restriction on projection signatures. The notation $\text{CHILD}_\mathcal{F}(\tau_1, \tau_2)$ means that $\tau_2$ is a child of $\tau_1$ in $(\rightarrow, \mathcal{F})$.

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

![Fig. 4. The projection signature graph of the example.](image-url)
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. \( \mathcal{DLR}_{ifd} \) [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].

5 Mapping \( \mathcal{DLR}^\pm \) to \( \mathcal{ALCQI} \)

We show that reasoning in \( \mathcal{DLR}^\pm \) is \textsc{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 \textsc{ExpTime}-complete [Baader et al., 2003].

We adapt and extend the mapping presented for \( \mathcal{DLR} \) knowledge bases in [Calvanese et al., 1998] 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].
induced by the projection signature.

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

$$\text{PATH}_\mathcal{S}(\tau, \tau')^\dagger = Q_{\tau_1} \circ \ldots \circ Q_{\tau_n} \circ Q_{\tau'}.$$ 

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 ($A_{R_i}^\tau$), for each relation name $R_i$ and each projected signature $\tau_j$ in the projection signature graph dominated by $\tau(R_i)$, while the edges are labelled by the roles $(Q_{\tau_j})$ needed for the reification.

The mapping $\cdot^\dagger$ is extended to concept and relation expressions as in Figure [6] with the proviso that whenever $\text{PATH}_\mathcal{S}(\tau_1, \tau_2)$ returns an empty path then the translation for
the corresponding expression becomes the bottom concept. Note that in $\mathcal{DLCR}^\pm$ the cardinals 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 $\exists \gamma [U_i] R_i$ with $q > 1$, then $U_i$ should not be mentioned in any other projection of the relation $R$ in such a way that $|\text{PATH}_{\mathcal{F}}(\tau(R), \{U_i\})| = 1$.

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 $\text{PATH}_{\mathcal{F}}(\tau(R_i), U_i)$ function. For example, considering Fig. 5, in order to access the attribute $\text{lobj}_p$ in Fig. 5, in order to access the attribute $\text{lobj}_p$, we extend the function $\gamma$ to singleton tuples with the meaning $\gamma_\text{lobj}(RN) \equiv |\text{lobj}_p| \leq 1$.

Clearly, the size of $\gamma(KB)$ is polynomial in the size of $KB$ (under the same coding of the numerical parameters), and thus we are able to state the main result of this paper.

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

**Proof.** We assume that the $KB$ is consistently rewritten by substituting each attribute with its canonical representative, thus, we do not have to deal with the renaming of attributes. Furthermore, we extend the function $\gamma$ to singleton tuples with the meaning $\gamma_\text{lobj}(RN) \equiv |\text{lobj}_p| \leq 1$.
that \( i(\langle U_1 : d_1 \rangle) = d_1 \).

(\Rightarrow) Let \( \mathcal{I} = (\Delta^\mathcal{I}, \mathcal{I}, \rho, i, \ell_{\mathcal{RN}}, \ldots) \) be a model for a \( \mathcal{DLR}^\pm \) knowledge base \( \mathcal{KB} \). To construct a model \( \mathcal{J} = (\Delta^\mathcal{J}, \mathcal{J}) \) for the \( \mathcal{ALCQI} \) knowledge base \( \gamma(\mathcal{KB}) \) we set \( \Delta^\mathcal{J} = \Delta^\mathcal{I} \). Furthermore, we set: \( (CN)^J = (CN)^I \), for every atomic concept \( CN \in C \), while for every \( \mathcal{RN} \in \mathcal{R} \) and \( \tau_i \in \mathcal{T}_{\mathcal{RN}} \) we set

\[
(A^\mathcal{J}_{\mathcal{RN}})^J = \{ i(\langle U_1 : d_1, \ldots U_k : d_k \rangle) \mid \{ U_1, \ldots, U_k \} = \tau_i \text{ and } \exists t \in \mathcal{R}^J, t[U_1] = d_1, \ldots, t[U_k] = d_k \}. \tag{2}
\]

For each role name \( Q_{\tau_i}, \tau_i \in \mathcal{J} \), we set

\[
(Q_{\tau_i})^J = \{ (d_1, d_2) \in \Delta^\mathcal{J} \times \Delta^\mathcal{J} \mid \exists t \in \mathcal{R}^J \text{ s.t. } d_1 = i(t[\tau_j]), d_2 = i(t[\tau_i]) \}
\text{ and CHILD}_{\mathcal{J}}(\tau_j, \tau_i), \text{ for some } \mathcal{RN} \in \mathcal{R} \}. \tag{3}
\]

For every \( \mathcal{RN} \in \mathcal{R} \) we set

\[
Q^J_{\mathcal{RN}} = \{ (d_1, d_2) \in \Delta^\mathcal{J} \times \Delta^\mathcal{J} \mid \exists t \in \mathcal{R}^J \text{ s.t. } d_1 = i(t) \text{ and } d_2 = \ell_{\mathcal{RN}}(t) \}, \tag{4}
\]

and

\[
(A^J_{\mathcal{RN}})^J = \{ \ell_{\mathcal{RN}}(t) \mid t \in \mathcal{R}^J \}. \tag{5}
\]

We now show that \( \mathcal{J} \) is indeed a model of \( \gamma(\mathcal{KB}) \).

1. \( \mathcal{J} \models \gamma_{d} \). This is a direct consequence of the fact that \( i \) is an injective function and that tuples with different arities are different tuples.

2. \( \mathcal{J} \models \gamma_{rel}(\mathcal{RN}) \), for every \( \mathcal{RN} \in \mathcal{R} \). We show that, for each \( \tau_i, \tau_j \text{ s.t. CHILD}_{\mathcal{J}}(\tau_i, \tau_j) \) and \( \tau_i \in \mathcal{T}_{\mathcal{RN}} \), \( \mathcal{J} \models A^J_{\mathcal{RN}} \subseteq \exists Q_{\tau_i}, A^J_{\mathcal{RN}} \) and \( \mathcal{J} \models \exists \geq \mathcal{Q}_{\tau_i}, \mathcal{T} \subseteq \bot \):
   - \( \mathcal{J} \models A^J_{\mathcal{RN}} \subseteq \exists Q_{\tau_i}, A^J_{\mathcal{RN}} \). Let \( d \in (A^J_{\mathcal{RN}})^J \). By (2), \( \exists t \in \mathcal{R}^J \text{ s.t. } d = i(t[\tau_j]) \).
     Since CHILD_{\mathcal{J}}(\tau_i, \tau_j), then \( \exists d' = i(t[\tau_j]) \) and, by (3), \( (d, d') \in Q^J_{\mathcal{RN}} \), while by (2), \( d' \in (A^J_{\mathcal{RN}})^J \). Thus, \( d \in (\exists Q_{\tau_i}, A^J_{\mathcal{RN}})^J \).
   - \( \mathcal{J} \models \exists \geq \mathcal{Q}_{\tau_i}, \mathcal{T} \subseteq \bot \). The fact that each \( Q_{\tau_i} \) is interpreted as a functional role is a direct consequence of the construction (3) and the fact that \( i \) is an injective function.

3. \( \mathcal{J} \models \gamma_{obj}(\mathcal{RN}) \), for every \( \mathcal{RN} \in \mathcal{R} \). Similar as above, considering the fact that each \( \ell_{\mathcal{RN}} \) is an injective function and equations (4)-(5).

4. \( \mathcal{J} \models C^J_{\mathcal{I}} \subseteq C^J_{\mathcal{I}} \) and \( \mathcal{J} \models R^J_{\mathcal{I}} \subseteq R^J_{\mathcal{I}} \). Since \( \mathcal{I} \models C^J_{\mathcal{I}} \subseteq C^J_{\mathcal{I}} \) and \( \mathcal{I} \models R^J_{\mathcal{I}} \subseteq R^J_{\mathcal{I}} \), it is enough to show the following:
   - \( d \in C^J \) iff \( d \in (C^J)^J \), for all \( \mathcal{DLR}^\pm \) concepts;
   - \( t \in R^J \) iff \( i(t) \in (R^J)^J \), for all \( \mathcal{DLR}^\pm \) relations.
Before we proceed with the proof, it is easy to show by structural induction that the following property holds:

\[
\text{If } i(t) \in R^I \text{ then } \exists t'(t') \in R^J \text{ s.t. } t = t'[\tau(R)] \text{, for some } \mathcal{RN} \in \mathcal{R}. \tag{6}
\]

We now proceed with the proof by structural induction. The base cases, for atomic concepts and roles, are immediate form the definition of both \( CN^J \) and \( \mathcal{RN}^J \). The
cases where complex concepts and relations are constructed using either boolean operators or global reification are easy to show. We thus show only the following cases.

Let \( d \in (\bigcup \mathcal{R})^J \). Then, \( d = \ell_{\mathcal{R}}(t) \) with \( t \in \mathcal{R}^J \). By induction, \( \nu(t) \in A_{\mathcal{R}}^J \) and, by \( \gamma_{\text{obj}}(\mathcal{R}) \), there is a \( d' \in \Delta^J \) s.t. \( \nu(t), d' \in Q_{\mathcal{R}}^J \) and \( d' \in (A_{\mathcal{R}}^J)^J \). By (4), \( d' = \ell_{\mathcal{R}}(t) \) and, since \( \ell_{\mathcal{R}} \) is injective, \( d' = d \). Thus, \( d \in (\bigcup \mathcal{R})^J \).

Let \( d \in (\exists^p q[U_i]^J)^J \). Then, there are different \( t_1, \ldots, t_q \in R^J \) s.t. \( t_i[U_i] = d, \) for all \( i = 1, \ldots, q \). By induction, \( \nu(t_i) \in R^J \) while, by (6), \( \nu(t'_i) \in R_{\mathcal{R}}^J \), for some atomic relation \( R_{\mathcal{R}} \in \mathcal{R} \) and a tuple \( t'_i \) s.t. \( t_i = t'_i[\tau(R)] \). By \( \gamma_{\text{rel}}(\mathcal{R}) \) and (3), \( \nu(t'_i), \nu(t_i) \in (\text{PATH}_{\mathcal{R}}(\tau(R), \{U_i\}))^J \) and \( \nu(t_i), d \in (\text{PATH}_{\mathcal{R}}(\tau(R), \{U_i\}))^J \).

Since \( \nu \) is injective, \( \nu(t_i) \neq \nu(t_j) \) when \( i \neq j \), thus, \( d \in (\exists^p q[U_i]^J)^J \).

Let \( t \in (\sigma_{U_i}, C)[R]^J \). Then, \( t \in R^J \) and \( t[U_i] \in C^J \), and, by induction, \( \nu(t) \in R^J \) and \( t[U_i] \in C^J \). As before, by \( \gamma_{\text{rel}}(\mathcal{R}) \) and by (3), \( \nu(t), t[U_i] \in (\text{PATH}_{\mathcal{R}}(\tau(R), \{U_i\}))^J \). Since \( \text{PATH}_{\mathcal{R}}(\tau(R), \{U_i\})^J \) is functional, then we have that \( \nu(t) \in (\sigma_{U_i}, C)[R]^J \).

Let \( t \in (\exists[U_1, \ldots, U_k]^J)^J \). Then, there is a tuple \( t' \in R^J \) s.t. \( t'[U_1, \ldots, U_k] = t \) and, by induction, \( \nu(t') \in R^J \). As before, by \( \gamma_{\text{rel}}(\mathcal{R}) \) and by (3) and (6), we can show that \( \nu(t'), \nu(t) \in (\text{PATH}_{\mathcal{R}}(\tau(R), \{U_1, \ldots, U_k\}))^J \) and thus \( \nu(t) \in (\exists[U_1, \ldots, U_k]^J)^J \).

All the other cases can be proved in a similar way. We now show the vice versa.

Let \( d \in (\bigcup \mathcal{R})^J \). Then, \( d \in (A_{\mathcal{R}}^J)^J \) and \( d = \ell_{\mathcal{R}}(t) \), for some \( t \in \mathcal{R}^J \), i.e., \( d \in (\bigcup \mathcal{R})^J \).

Let \( d \in (\exists^p q[U_i]^J)^J \). Then, there are different \( d_1, \ldots, d_q \in \Delta^J \) s.t. \( d_l, d \in (\text{PATH}_{\mathcal{R}}(\tau(R), \{U_i\}))^J \) and \( d_l \in R^J \), for \( l = 1, \ldots, q \). By induction, each \( d_l = d_l \) and \( t_l \in R^J \). Since \( \nu \) is injective, then \( t_l \neq t_j \) for all \( l, j = 1, \ldots, q, l \neq j \).

We need to show that \( t_l[U_i] = d, \) for all \( l = 1, \ldots, q \). By (3) and the fact that \( (d_l, d) \in (\text{PATH}_{\mathcal{R}}(\tau(R), \{U_i\}))^J, \) then \( d = d[t_l[U_i]] = t_l[U_i] \).

Let \( \nu(t) \in (\sigma_{U_i}, C)[R]^J \). Then, \( \nu(t) \in R^J \) and, by induction, \( t \in R^J \). Let \( t[U_i] = d \). We need to show that \( d \in C^J \). By \( \gamma_{\text{rel}}(\mathcal{R}) \) and by (3) and (6), \( \nu(t), d \in (\text{PATH}_{\mathcal{R}}(\tau(R), \{U_i\}))^J \), then \( d \in C^J \) and, by induction, \( d \in C^J \).

Let \( \nu(t) \in (\exists[U_1, \ldots, U_k]^J)^J \). Then, there is \( d \in \Delta^J \) s.t.

\[
(\exists d, \nu(t) \in (\text{PATH}_{\mathcal{R}}(\tau(R), \{U_1, \ldots, U_k\}))^J \land d \in R^J)
\]

and \( d \in R^J \). By induction, \( d = \nu(t') \) and \( t' \in R^J \). By (3), \( \nu(t) = \nu(t'[U_1, \ldots, U_k]) \), i.e., \( t = t'[U_1, \ldots, U_k] \). Thus, \( t \in (\exists[U_1, \ldots, U_k]^J)^J \).

\((\Leftarrow)\) Let \( \mathcal{J} = (\Delta^J, \rho, \ell_{\mathcal{R}}, \ldots) \) for a \( \mathcal{DCR} \) knowledge base \( \mathcal{KB} \). We set: \( \Delta^J = \Delta^J, \) \( CN^J = (CN^J)^J \), for every atomic concept \( CN \in \mathcal{C} \), while, for every \( R \in \mathcal{R} \), we set:

\[
\mathcal{R}^J = \{ t = \langle U_1 : d_1, \ldots, U_n : d_n \rangle \in T_{\Delta^J}(\tau(\mathcal{R})) \mid \exists d \in A_{\mathcal{R}}^J \text{ s.t.}
\]

\[
(d, t[U_i]) \in (\text{PATH}_{\mathcal{R}}(\tau(\mathcal{R}), \{U_i\}))^J \text{ for } i = 1, \ldots, n. \} \quad (7)
\]

Since \( \mathcal{J} \) satisfies \( \gamma_{\text{rel}}(\mathcal{R}) \), then, for every \( d \in A_{\mathcal{R}}^J \), there is a unique tuple \( \langle U_1 : d_1, \ldots, U_n : d_n \rangle \in \mathcal{R}^J \), we say that \( d \) generates \( \langle U_1 : d_1, \ldots, U_n : d_n \rangle \) and, in
symbols, \( d \rightarrow \langle U_1 : d_1, \ldots, U_n : d_n \rangle \). Furthermore, since \( \mathcal{J} \) is tree shaped, to each

tuple corresponds a unique \( d \) that generates it. Thus, let \( d \rightarrow \langle U_1 : d_1, \ldots, U_n : d_n \rangle \), by setting

\[ i(\langle U_1 : d_1, \ldots, U_n : d_n \rangle) = d, \quad \text{s.t.} \quad (d, d_{\tau_i}) \in (\text{PATH}_{\mathcal{J}}(\{U_1, \ldots, U_n\}, \tau_i))^{\mathcal{J}}, \]  

(8)

for all \( \tau_i \in \mathcal{J}(U_1, \ldots, U_n) \), then the function \( i \) is as required.

By setting

\[
\ell_{RN}(\langle U_1 : d_1, \ldots, U_n : d_n \rangle) = d, \quad \text{s.t}.
\]

\[ (i(\langle U_1 : d_1, \ldots, U_n : d_n \rangle), d) \in Q_{RN}^\mathcal{J}, \]  

(9)

by \( \gamma_{\text{obj}}(RN) \), both \( Q_{RN} \) and its inverse are interpreted as a functional roles by \( \mathcal{J} \),

Thus the function \( \ell_{RN} \) is as required.

It is easy to show by structural induction that the following property holds:

If \( t \in R^T \) then \( \exists t' \in RN^T \) s.t. \( t = t'_{\uparrow}([\tau(R)]) \), for some \( RN \in \mathcal{R} \).

(10)

We now show that \( \mathcal{I} \) is indeed a model of \( KB \), i.e., \( \mathcal{I} \models C_1 \subseteq C_2 \) and \( \mathcal{I} \models R_1 \subseteq R_2 \). As before, since \( \mathcal{J} \models C_1 \subseteq C_2 \) and \( \mathcal{J} \models R_1 \subseteq R_2 \), it is easy to show the following:

- \( d \in C^\mathcal{J} \) iff \( d \in (C^\downarrow)^{\mathcal{J}} \), for all \( \mathcal{DLR}^\pm \) concepts;
- \( t \in R^\mathcal{J} \) iff \( \langle t \rangle \in (R^\downarrow)^{\mathcal{J}} \), for all \( \mathcal{DLR}^\pm \) relations.

The proof is by structural induction. The base cases are trivially true. Similarly for the boolean operators and global reification. We thus show only the following cases.

Let \( d \in (\bigcirc RN)^T \). Then, \( d = \ell_{RN}(t) \) with \( t \in RN^T \). By induction, \( \langle t \rangle \in A_{RN}^\mathcal{J} \),

and, by \( \gamma_{\text{obj}}(RN) \), there is a \( d' \in \Delta^\mathcal{J} \) s.t. \( \langle t \rangle \in Q_{RN}^{\mathcal{J}} \) and \( d' \in (A_{RN}^{\mathcal{J}})^{\mathcal{J}} \).

By (2), \( d = d' \) and thus, \( d \in (\bigcirc RN)^{\mathcal{J}} \).

Let \( d \in (\exists \emptyset\emptyset q[U_1]R)^T \). Then, there are different \( t_1, \ldots, t_q \in R^T \) s.t. \( t[U_1] = d \),

for all \( l = 1, \ldots, q \). For each \( t_j \), by (10), there is a \( t_j' \in RN^T \) s.t. \( t_j'_{\uparrow}([\tau(R)]) \),

for some \( RN \in \mathcal{R} \), while, by induction, \( \langle t_j \rangle \in R_{\uparrow}^\mathcal{J} \) and \( \langle t_j' \rangle \in RN_{\uparrow}^{\mathcal{J}} \). Thus,

\( \exists \emptyset\emptyset q[U_1]R \sigma_{U_j,C}R \). Then, \( t \in R^T \) and \( t[U_1] = d \in C^\mathcal{J} \). By induction, \( \langle t \rangle \in R_{\uparrow}^\mathcal{J} \) and \( d \in C^\mathcal{J} \).

As before, by (7), (8) and (10), we can show that \( \langle t \rangle \in (\text{PATH}_\mathcal{J}(\tau(R), \{U_1\}) \sigma_{U_j,C}R)^{\mathcal{J}} \), and, since \( \text{PATH}_\mathcal{J}(\tau(R), \{U_1\}) \) is functional, then \( \langle t \rangle \in (\sigma_{U_j,C}R)^{\mathcal{J}} \).

Let \( t \in (\exists[U_1, \ldots, U_k]R)^T \). Then, there is a tuple \( t' \in R^T \) s.t. \( t'[U_1, \ldots, U_k] = t \)
and, by induction, \( \iota(t') \in R_1^{\mathcal{I}} \). As before, by (8) and (10), we can show that 
\((\iota(t'), \iota(t)) \in \text{PATH}_\mathcal{I}(\tau(R), \{U_1, \ldots, U_k\})^{\mathcal{I}} \) and thus \( \iota(t) \in (\exists[\iota(U_1), \ldots, \iota(U_k)]^\mathcal{I}) \).

All the other cases can be proved in a similar way. We now show the vice versa.

Let \( d \in (\bigotimes \text{RN})^{\mathcal{I}} \). Then, \( d \in (A_1^{\text{ RN}})^{\mathcal{I}} \) and, by \( \gamma_{\text{lbox}}(\text{ RN}) \), there is a \( d' \in \Delta^{\mathcal{I}} \) s.t. \((d', d) \in Q^{\mathcal{I}}_\text{RN} \) and \( d' \in A_1^{\text{ RN}} \). By induction, \( d' = \iota(t') \) with \( t' \in \text{RN}^{\mathcal{I}} \) and thus, 
\((\iota(t'), d) \in Q^{\mathcal{I}}_\text{RN} \) and, by (9), \( \ell^{\text{ RN}}(t') = d \), i.e., \( d \in (\bigotimes \text{ RN})^{\mathcal{I}} \).

Let \( d \in (\exists^{\mathcal{I}}[U_i])^{\mathcal{I}} \). Thus, there are different \( d_1, \ldots, d_q \in \Delta^{\mathcal{I}} \) s.t. \((d_i, d) \in (\text{PATH}_\mathcal{I}(\tau(R), \{U_i\})^{\mathcal{I}} \) and \( d_i \in \text{RN}^{\mathcal{I}} \), for \( i = 1, \ldots, q \). By induction, each \( d_i = \iota(t_i) \) and \( t_i \in \text{RN}^{\mathcal{I}} \). Since \( \iota \) is injective, then \( t_i \neq t_j \) for all \( i, j = 1, \ldots, q \), \( l \neq j \). We need to show that \( t_i[U_i] = d \), for all \( i = 1, \ldots, q \). By (10), there is a \( t'_i \in \text{RN}^{\mathcal{I}} \) s.t. \( t_i = t'_i[\tau(R)] \), for some \( RN \in \mathcal{R} \) and, by (8), \( (\iota(t'_i), \iota(t_i)) \in (\text{PATH}_\mathcal{I}(\tau(R), \{U_i\})^{\mathcal{I}} \). Since \( (\iota(t_i), d) \in (\text{PATH}_\mathcal{I}(\tau(R), \{U_i\})^{\mathcal{I}} \) and \( \text{PATH}_\mathcal{I} \) is functional in \( \mathcal{DLR} \), then, \( (\iota(t'_i), d) \in (\text{PATH}_\mathcal{I}(\tau(RN), \{U_i\})^{\mathcal{I}} \) and, by (7), \( t'_i[U_i] = t_i[U_i] = d \).

Let \( \iota(t) \in (\sigma_{U_i:C}R)^{\mathcal{I}} \). Thus, \( \iota(t) \in R_1^{\mathcal{I}} \), and, by induction, \( t \in \text{RN}^{\mathcal{I}} \). Let \( t[U_i] = d \). We need to show that \( d \in C^\mathcal{I} \). As before, by (10) and (8), we have that \( \iota(t), d) \in (\text{PATH}_\mathcal{I}(\tau(R), \{U_i\})^{\mathcal{I}} \). Then \( d \in C^\mathcal{I} \) s.t. and, by induction, \( d \in C^\mathcal{I} \).

Let \( \iota(t) \in (\exists[\iota(U_1), \ldots, \iota(U_k)]^\mathcal{I}) \). Then, there is \( d \in \Delta^{\mathcal{I}} \) s.t.
\[
(d, \iota(t)) \in (\text{PATH}_\mathcal{I}(\tau(R), \{U_1, \ldots, U_k\})^{\mathcal{I}} \]

and \( d \in R_1^{\mathcal{I}} \). By induction, \( d = \iota(t') \) and \( t' \in \text{RN}^{\mathcal{I}} \). As before, by (8) and (10), we can show that there is a tuple \( t'' \in \text{RN} \) s.t. \( (\iota(t''), \iota(t)) \in (\text{PATH}_\mathcal{I}(\tau(RN), \{U_1, \ldots, U_k\})^{\mathcal{I}} \) and thus, \( t = (\iota(U_1, \ldots, U_k), \ldots, (\exists[\iota(U_1), \ldots, \iota(U_k)]^\mathcal{I}) \).

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

**Corollary 3.** Reasoning in \( \mathcal{DLR}^\mathcal{I} \) 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 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 Computing*, 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.