# The $\mathcal{DLR}_{\mathcal{US}}$ Temporal Description Logic


Alessandro Artale
Dept. of Computation
UMIST, Manchester, UK
artale@co.umist.ac.uk

Enrico Franconi
Dept. of Computer Science
Univ. of Manchester, UK
franconi@cs.man.ac.uk

Milenko Mosurović
Univ. of Montenegro, YU
milenko@cg.ac.yu

Frank Wolter
Inst. für Informatik
Univ. of Leipzig, D
wolter@informatik.uni-leipzig.de

Michael Zakharyaschev
Dept. of Computer Science
Kings College, London, UK
mz@dcs.kcl.ac.uk



**Abstract**

This paper introduces a new logical formalism, intended for temporal conceptual modelling, as a natural combination of the well-known description logic $\mathcal{DLR}$ and point-based linear temporal logic with *Since* and *Until*. We define a query language (where queries are non-recursive Datalog programs and atoms are complex $\mathcal{DLR}_{\mathcal{US}}$ expressions) and investigate the problem of checking query containment under the constraints defined by $\mathcal{DLR}_{\mathcal{US}}$ conceptual schemas, as well as the problems of schema satisfiability and logical implication. Although it is shown that reasoning in full $\mathcal{DLR}_{\mathcal{US}}$ is undecidable, we identify the decidable (in a sense, maximal) fragment $\mathcal{DLR}_{\mathcal{US}}^{-}$ by allowing applications of temporal operators to formulas and entities only (but not to relation expressions). We obtain the following hierarchy of complexity results: (a) reasoning in $\mathcal{DLR}_{\mathcal{US}}^{-}$ with atomic formulas is EXPTIME-complete, (b) satisfiability and logical implication of arbitrary $\mathcal{DLR}_{\mathcal{US}}^{-}$ formulas is EXPSPACE-complete, and (c) the problem of checking query containment of non-recursive Datalog queries under $\mathcal{DLR}_{\mathcal{US}}^{-}$ constraints is decidable in 2EXPTIME.

# 1 Introduction

The *temporal description logic* $\mathcal{DLR}_{\mathcal{US}}$ we design in this paper is based on the expressive and decidable description logic $\mathcal{DLR}$ which allows the logical reconstruction and the extension of representational tools such as object-oriented data models (e.g., class diagrams in UML and ODMG), semantic data models (e.g., extended entity-relationship, EER, and ORM), frame-based ontology languages (e.g., OKBC, XOL, and OIL), and semantic networks [6, 7]. In this setting, an interesting feature of $\mathcal{DLR}$ is the ability to completely *define* entities and relations as $\mathcal{DLR}$ views over other entities and relations of the conceptual schema. Moreover, $\mathcal{DLR}$ formulas can express a large class of integrity constraints that are typical in databases, for instance, existence dependencies, exclusion dependencies, typed inclusion dependencies without projection of relations, unary inclusion dependencies, full key dependencies [5]. Logical implication in $\mathcal{DLR}$ is EXPTIME-complete [5]; efficient, correct and complete algorithms exist in implemented systems which are used in real applications [16, 17, 11].

$\mathcal{DLR}$ is not only a very powerful language for conceptual data modelling. The problem of view-based query processing under $\mathcal{DLR}$ constraints has also been studied [5]. Checking query containment of non-recursive Datalog queries under $\mathcal{DLR}$ constraints is decidable in 2EXPTIME [5, 15].

Given all these nice features of $\mathcal{DLR}$, it is natural to try to extend it with a temporal dimension, to understand the expressive power of the resulting hybrid with respect to the needs of temporal conceptual modelling and view based query processing, and to investigate its computational properties. This paper reports the results of such an attempt.

We construct $\mathcal{DLR}_{\mathcal{US}}$ as an organic combination of $\mathcal{DLR}$ and the propositional linear temporal logic with *Since* and *Until* [21, 12] (which usually serves as the temporal component in the first-order approach) by allowing applications of temporal operators to all syntactical terms of $\mathcal{DLR}$: entities, relations, and schemas. Previous approaches to temporal description logics considered much weaker languages in the tradition of description logics having only binary relations (i.e., roles) [20, 24, 27]. For the $\mathcal{ALC}$ fragment a tableau-based algorithm has been studied in [23]. For a survey of various approaches to temporal description logics see [3].

To illustrate the expressive power of $\mathcal{DLR}_{\mathcal{US}}$, we have provided a formal semantic characterisation—by means of $\mathcal{DLR}_{\mathcal{US}}$ theories—of the most important temporal conceptual modelling constructs (for the valid time representation) appeared in the literature on the entity-relationship data model [13, 22, 2]. To the best of our knowledge, this is the first systematic formalisation of the constructs present in the majority of temporal conceptual modelling systems. The outcome is an elegant correspondence between temporal constructs and sets of $\mathcal{DLR}_{\mathcal{US}}$ formulas. Moreover, temporal integrity constraints can be captured by additional $\mathcal{DLR}_{\mathcal{US}}$ formulas. In this paper we do not go into the details of this part; these can be found in [4].

We then investigate computational properties of reasoning with $\mathcal{DLR}_{\mathcal{US}}$ by analysing schema, entity, and relation satisfiability, logical implication, and query containment for non-recursive Datalog queries under constraints imposed by $\mathcal{DLR}_{\mathcal{US}}$ conceptual schemas.

The full $\mathcal{DLR}_{\mathcal{US}}$ turns out to be undecidable. The main reason for this is the possibility to postulate that a binary relation does not vary in time—a very small fragment of $\mathcal{DLR}$ (say, the basic description logic $\mathcal{ALC}$) can encode then the undecidable tiling problem (cf. [25, 14]). The fragment $\mathcal{DLR}^{-}_{\mathcal{US}}$ of $\mathcal{DLR}_{\mathcal{US}}$ deprived of this ability to talk about temporal persistence of $n$-ary relations, for $n \geq 2$, is still very expressive, as is illustrated by examples provided in [4], but its computational behaviour is much better. We obtain the following hierarchy of complexity results: (1) reasoning in $\mathcal{DLR}^{-}_{\mathcal{US}}$ with atomic formulas is EXPTIME-complete, (2) satisfiability and logical implication of arbitrary $\mathcal{DLR}^{-}_{\mathcal{US}}$ formulas is EXPSPACE-complete, and (3) the problem of checking query containment of non-recursive Datalog queries under $\mathcal{DLR}^{-}_{\mathcal{US}}$ constraints is decidable in 2EXPTIME. The latter result is the first decidability result we are aware of on containment of temporal conjunctive queries under expressive constraints

The paper is organised as follows. Section 2 introduces the syntax and the semantics of $\mathcal{DLR}_{\mathcal{US}}$, and provides a worked out example. In Section 5 decidability and complexity results are summarised for satisfiability and logical implication in $\mathcal{DLR}_{\mathcal{US}}$ and $\mathcal{DLR}^{-}_{\mathcal{US}}$, as well as for the query containment problem.

# 2 The Temporal Description Logic

In this paper, we adopt the *snapshot* representation of abstract temporal databases (and temporal conceptual models); see e.g. [10]. The flow of time $\mathcal{T} = \langle \mathcal{T}_p, < \rangle$, where $\mathcal{T}_p$ is a set of time points (or chronons) and $<$ a binary precedence relation on $\mathcal{T}_p$, is assumed to be isomorphic to $\langle \mathbb{Z}, < \rangle$. Thus, a temporal database can be regarded as a map from time points in $\mathcal{T}$ to standard (relational) databases with the same domain of attributes and the same interpretation of constants.

As a language of temporal database conceptual schemas we use a natural combination of the propositional linear temporal logic with *Since* and *Until* [21, 12] and the (non-temporal) description logic $\mathcal{DLR}$ [5]. The resulting *temporal description logic* will be denoted by $\mathcal{DLR}_{\mathcal{US}}$.

The basic syntactical types of $\mathcal{DLR}_{\mathcal{US}}$ are *entities* (i.e., unary predicates, also known as *concepts*) and $n$-ary *relations* of arity $\geq 2$. Starting from a set $EN$ of *atomic entities* and a set $RN$ of *atomic relations* we define inductively (complex) entity and relation expressions as is shown in the upper part of Fig. 1, where the binary constructs ($\sqcap, \sqcup, \mathcal{U}, \mathcal{S}$) are applied to relations of the same arity, $i$, $j$, $k$, $n$ are natural numbers, $i \leq n$, and $j$ does not exceed the arity of $R$.

A *temporal conceptual database schema* (or a *knowledge base*) is a finite set $\Sigma$ of $\mathcal{DLR}_{\mathcal{US}}$-formulas. Atomic formulas are formulas of the form $E_1 \sqsubseteq E_2$ and $R_1 \sqsubseteq R_2$,

$$
\begin{array}{rcl}
R & \rightarrow & \top_n \mid RN \mid \neg R \mid R_1 \sqcap R_2 \mid R_1 \sqcup R_2 \mid i/n : E \mid \\
 & & \Diamond^+ R \mid \Diamond^- R \mid \Box^+ R \mid \Box^- R \mid \oplus R \mid \ominus R \mid R_1 \,\mathcal{U}\, R_2 \mid R_1 \,\mathcal{S}\, R_2 \\
E & \rightarrow & \top \mid EN \mid \neg E \mid E_1 \sqcap E_2 \mid E_1 \sqcup E_2 \mid \exists^{\leqslant k}[j]R \mid \\
 & & \Diamond^+ E \mid \Diamond^- E \mid \Box^+ E \mid \Box^- E \mid \oplus E \mid \ominus E \mid E_1 \,\mathcal{U}\, E_2 \mid E_1 \,\mathcal{S}\, E_2
\end{array}
$$

$$
\begin{array}{rcl}
(\top_n)^{\mathcal{I}(t)} & \subseteq & (\Delta^{\mathcal{I}})^n \\
RN^{\mathcal{I}(t)} & \subseteq & (\top_n)^{\mathcal{I}(t)} \\
(\neg R)^{\mathcal{I}(t)} & = & (\top_n)^{\mathcal{I}(t)} \setminus R^{\mathcal{I}(t)} \\
(R_1 \sqcap R_2)^{\mathcal{I}(t)} & = & R_1^{\mathcal{I}(t)} \cap R_2^{\mathcal{I}(t)} \\
(i/n : E)^{\mathcal{I}(t)} & = & \{ \langle d_1, \ldots, d_n \rangle \in (\top_n)^{\mathcal{I}(t)} \mid d_i \in E^{\mathcal{I}(t)} \} \\
(R_1 \,\mathcal{U}\, R_2)^{\mathcal{I}(t)} & = & \{ \langle d_1, \ldots, d_n \rangle \in (\top_n)^{\mathcal{I}(t)} \mid \exists v > t. (\langle d_1, \ldots, d_n \rangle \in R_2^{\mathcal{I}(v)} \wedge \\
 & & \forall w \in (t, v). \langle d_1, \ldots, d_n \rangle \in R_1^{\mathcal{I}(w)}) \} \\
(R_1 \,\mathcal{S}\, R_2)^{\mathcal{I}(t)} & = & \{ \langle d_1, \ldots, d_n \rangle \in (\top_n)^{\mathcal{I}(t)} \mid \exists v < t. (\langle d_1, \ldots, d_n \rangle \in R_2^{\mathcal{I}(v)} \wedge \\
 & & \forall w \in (v, y). \langle d_1, \ldots, d_n \rangle \in R_1^{\mathcal{I}(w)}) \} \\
(\Diamond^+ R)^{\mathcal{I}(t)} & = & \{ \langle d_1, \ldots, d_n \rangle \in (\top_n)^{\mathcal{I}(t)} \mid \exists v > t. \langle d_1, \ldots, d_n \rangle \in R^{\mathcal{I}(v)} \} \\
(\oplus R)^{\mathcal{I}(t)} & = & \{ \langle d_1, \ldots, d_n \rangle \in (\top_n)^{\mathcal{I}(t)} \mid \langle d_1, \ldots, d_n \rangle \in R^{\mathcal{I}(t+1)} \} \\
(\Diamond^- R)^{\mathcal{I}(t)} & = & \{ \langle d_1, \ldots, d_n \rangle \in (\top_n)^{\mathcal{I}(t)} \mid \exists v < t. \langle d_1, \ldots, d_n \rangle \in R^{\mathcal{I}(v)} \} \\
(\ominus R)^{\mathcal{I}(t)} & = & \{ \langle d_1, \ldots, d_n \rangle \in (\top_n)^{\mathcal{I}(t)} \mid \langle d_1, \ldots, d_n \rangle \in R^{\mathcal{I}(t-1)} \} \\
\top^{\mathcal{I}(t)} & = & \Delta^{\mathcal{I}} \\
EN^{\mathcal{I}(t)} & \subseteq & \top^{\mathcal{I}(t)} \\
(\neg E)^{\mathcal{I}(t)} & = & \top^{\mathcal{I}(t)} \setminus E^{\mathcal{I}(t)} \\
(E_1 \sqcap E_2)^{\mathcal{I}(t)} & = & E_1^{\mathcal{I}(t)} \cap E_2^{\mathcal{I}(t)} \\
(\exists^{\leqslant k}[j]R)^{\mathcal{I}(t)} & = & \{ d \in \top^{\mathcal{I}(t)} \mid \sharp\{ \langle d_1, \ldots, d_n \rangle \in R^{\mathcal{I}(t)} \mid d_j = d \} \leqslant k \} \\
(E_1 \,\mathcal{U}\, E_2)^{\mathcal{I}(t)} & = & \{ d \in \top^{\mathcal{I}(t)} \mid \exists v > t. (d \in E_2^{\mathcal{I}(v)} \wedge \forall w \in (t, v). d \in E_1^{\mathcal{I}(w)}) \} \\
(E_1 \,\mathcal{S}\, E_2)^{\mathcal{I}(t)} & = & \{ d \in \top^{\mathcal{I}(t)} \mid \exists v < t. (d \in E_2^{\mathcal{I}(v)} \wedge \forall w \in (v, t). d \in E_1^{\mathcal{I}(w)}) \}
\end{array}
$$

Figure 1: Syntax and semantics of $\mathcal{DLR}_{\mathcal{US}}$.

with $R_1$ and $R_2$ being relations of the same arity. If $\varphi$ and $\psi$ are $\mathcal{DLR}_{\mathcal{US}}$-formulas, then so are $\neg\varphi, \varphi \wedge \psi, \varphi \mathcal{U} \psi, \varphi \mathcal{S} \psi$. $E_1 \doteq E2$ is used as an abbreviation for $(E_1 \sqsubseteq E_2) \wedge (E_2 \sqsubseteq E_1)$, for both entities and relations. Temporal conceptual database schemas will serve as constraints for temporal databases.

The language of $\mathcal{DLR}_{\mathcal{US}}$ is interpreted in *temporal models* over $\mathcal{T}$, which are triples of the form $\mathcal{I} \doteq \langle \mathcal{T}, \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}(t)} \rangle$, where $\Delta^{\mathcal{I}}$ is non-empty set of objects (the *domain* of $\mathcal{I}$) and $\cdot^{\mathcal{I}(t)}$ an *interpretation function* such that, for every $t \in \mathcal{T}$, every entity $E$, and every $n$-ary relation $R$, we have $E^{\mathcal{I}(t)} \subseteq \Delta^{\mathcal{I}}$ and $R^{\mathcal{I}(t)} \subseteq (\Delta^{\mathcal{I}})^n$. The semantics of entity and relation expressions is defined in the lower part of Fig. 1, where $(u, v) = \{ w \in \mathcal{T} \mid u < w < v \}$ and the operators $\Box^+$ (always in the future) and $\Box^-$ (always in the past) are the duals of $\Diamond^+$ (some time in the future) and $\Diamond^-$

(some time in the past), respectively, i.e., $\Box^+ E \equiv \neg\Diamond^+\neg E$ and $\Box^- E \equiv \neg\Diamond^-\neg E$, for both entities and relations. For entities, the temporal operators $\Diamond^+$, $\oplus$ (at the next moment), and their past counterparts can be defined via $\mathcal{U}$ and $\mathcal{S}$: $\Diamond^+ E \equiv \top\,\mathcal{U}\,E$, $\oplus\, E \equiv \bot\,\mathcal{U}\,E$, etc. However, this is not possible for relations of arity $> 1$, since $\top_n$—the top $n$-ary relation—can be interpreted by different subsets of the $n$-ary cross product $\top \times \cdots \times \top$ at different time points.[1] The operators $\Diamond^*$ (at some moment) and its dual $\Box^*$ (at all moments) can be defined for both entities and relations as $\Diamond^* E \equiv E \sqcup \Diamond^+ E \sqcup \Diamond^- E$ and $\Box^* E \equiv E \sqcap \Box^+ E \sqcap \Box^- E$, respectively.

The non-temporal fragment of $\mathcal{DLR}_{\mathcal{US}}$ coincides with $\mathcal{DLR}$. For both entity and relation expressions all the Boolean constructs are available. The selection expression $i/n : E$ denotes an $n$-ary relation whose $i$th argument ($i \leq n$) is of type $E$; if it is clear from the context, we omit $n$ and write $(i : E)$. The projection expression $\exists^{\lessgtr k}[i]R$ is a generalisation with cardinalities of the projection operator over the $i$th argument of the relation $R$ (which coincides with $\exists^{\geq 1}[i]R$). It is also possible to use the named attribute version of the model by replacing argument position numbers with *role names*.

Given a formula $\varphi$, an interpretation $\mathcal{I}$, and a time point $t \in \mathcal{T}$, the truth-relation $\mathcal{I}, t \models \varphi$ ($\varphi$ holds in $\mathcal{I}$ at moment $t$) is defined inductively as follows:

$$
\begin{array}{ll}
\mathcal{I}, t \models E_1 \sqsubseteq E_2 & \text{iff } E_1^{\mathcal{I}(t)} \subseteq E_2^{\mathcal{I}(t)} \\
\mathcal{I}, t \models R_1 \sqsubseteq R_2 & \text{iff } R_1^{\mathcal{I}(t)} \subseteq R_2^{\mathcal{I}(t)} \\
\mathcal{I}, t \models \varphi \wedge \psi & \text{iff } \mathcal{I}, t \models \varphi \text{ and } \mathcal{I}, t \models \psi \\
\mathcal{I}, t \models \neg\varphi & \text{iff } \mathcal{I}, t \not\models \varphi \\
\mathcal{I}, t \models \varphi\,\mathcal{U}\,\psi & \text{iff } \exists v > t.(\mathcal{I}, v \models \psi \wedge \forall w \in (t, v).\mathcal{I}, w \models \varphi) \\
\mathcal{I}, t \models \varphi\,\mathcal{S}\,\psi & \text{iff } \exists v < t.(\mathcal{I}, v \models \psi \wedge \forall w \in (v, t).\mathcal{I}, w \models \varphi)
\end{array}
$$

A formula $\varphi$ is called *satisfiable* if there is a temporal model $\mathcal{I}$ such that $\mathcal{I}, t \models \varphi$, for some time point $t$. A conceptual schema $\Sigma$ is *satisfiable* if the conjunction $\bigwedge \Sigma$ of all formulas in $\Sigma$ is satisfiable (we write $\mathcal{I}, t \models \Sigma$ instead of $\mathcal{I}, t \models \bigwedge \Sigma$); in this case $\mathcal{I}$ is called a *model* of $\Sigma$. We say that $\Sigma$ is *globally satisfiable* if there is $\mathcal{I}$ such that $\mathcal{I}, t \models \Sigma$ for every $t$ ($\mathcal{I} \models \Sigma$, in symbols). An entity $E$ (or relation $R$) is *satisfiable* if there is $\mathcal{I}$ such that $E^{\mathcal{I}(t)} \neq \emptyset$ (respectively, $R^{\mathcal{I}(t)} \neq \emptyset$), for some time point $t$. Finally, we say that $\Sigma$ (*globally*) *implies* $\varphi$ and write $\Sigma \models \varphi$ if we have $\mathcal{I} \models \varphi$ whenever $\mathcal{I} \models \Sigma$.

Note that an entity $E$ is satisfiable iff $\neg(E \sqsubseteq \bot)$ is satisfiable. An $n$-ary relation $R$ is satisfiable iff $\neg(\exists^{\geq 1}[i]R \sqsubseteq \bot)$ is satisfiable for some $i \leq n$. A conceptual schema $\Sigma$ is globally satisfiable iff $\Box^*(\bigwedge \Sigma)$ is satisfiable. And $\Sigma \models \varphi$ iff $\Box^*(\bigwedge \Sigma) \wedge \neg\varphi$ is not satisfiable. Thus, all reasoning tasks connected with the notions introduced above reduce to satisfiability of formulas.

The logic $\mathcal{DLR}_{\mathcal{US}}$ can be regarded as a rather expressive fragment of the first-order temporal logic $L^{\{\textbf{since, until}\}}$; cf. [10, 14] and Section 5 below.

[1] For instance, we may have $\langle d_1, d_2\rangle \in (\Diamond^+ R)^{\mathcal{I}(t)}$ because $\langle d_1, d_2\rangle \in R^{\mathcal{I}(t+2)}$, but $\langle d_1, d_2\rangle \notin (\top_2)^{\mathcal{I}(t+1)}$.

# 3 Temporal queries

One more important reasoning task is known as the problem of query containment (see, e.g., [10, 8, 1] for a survey and a discussion about temporal queries). A *non-recursive Datalog query* (i.e., a disjunction of conjunctive queries or SPJ-queries) over a $\mathcal{DLR}_{\mathcal{US}}$ schema $\Sigma$ is an expression of the form

$$\mathcal{Q}(\overrightarrow{x}) \mathrel{:-} \bigvee_j \mathcal{Q}_j(\overrightarrow{x}, \overrightarrow{y_j}, \overrightarrow{c_j}),$$

where each $\mathcal{Q}_j$ is a conjunction of atoms

$$\mathcal{Q}_j(\overrightarrow{x}, \overrightarrow{y_j}, \overrightarrow{c_j}) \equiv \bigwedge_i P_j^i(\overrightarrow{x}_j^i, \overrightarrow{y}_j^i, \overrightarrow{c}_j^i),$$

$P_j^i$ are $\mathcal{DLR}_{\mathcal{US}}$ entity or relation expressions, $\overrightarrow{x}_j^i$, $\overrightarrow{y}_j^i$, and $\overrightarrow{c}_j^i$ are sequences of distinguished variables, existential variables, and constants, respectively, the number of which is in agreement with the arity of $P_j^i$. The variables $\overrightarrow{x}$ in the head are the union of all the distinguished variables in each $\mathcal{Q}_j$; the existential variables are used to make coreferences in the query, and constants are fixed values. The arity of $\mathcal{Q}$ is the number of variables in $\overrightarrow{x}$.

It is to be noted that we allow entities and relations in the query to occur in the conceptual schema $\Sigma$. This approach is similar to that of [5], where atoms in a query can be constrained by means of schema formulas.

The semantics of queries is defined as follows. Let $\mathcal{I}$ be a temporal model and $t$ a time point in $\mathcal{T}$ such that $\mathcal{I}$ satisfies $\Sigma$ at $t$, i.e., $\mathcal{I}, t \models \Sigma$. The snapshot interpretation

$$\mathcal{I}(t) = \left\langle \Delta^{\mathcal{I}}, \{E^{\mathcal{I}(t)} \mid E \in EN\}, \{R^{\mathcal{I}(t)} \mid R \in RN\} \right\rangle$$

can be regarded as a usual first-order structure (i.e., a snapshot non-temporal database at time $t$ conforming in a sense to the conceptual schema), and so the whole $\mathcal{I}$ as a first-order temporal model (with constant domain $\Delta^{\mathcal{I}}$ in which some values of the query constants are specified). The *evaluation* of a query $\mathcal{Q}$ of arity $n$ under the constraints $\Sigma$ in the model $\mathcal{I}$ at moment $t$ is the set

$$\mathsf{ans}(\mathcal{Q}, \mathcal{I}(t)) = \{\overrightarrow{o} \in (\Delta^{\mathcal{I}})^n \mid \mathcal{I}, t \models \bigvee_j \exists \overrightarrow{y_j}. \mathcal{Q}_j(\overrightarrow{o}, \overrightarrow{y_j}, \overrightarrow{c_j})\}$$

Given two queries (of the same arity) $\mathcal{Q}_1$ and $\mathcal{Q}_2$ over $\Sigma$, we say that $\mathcal{Q}_1$ is *contained* in $\mathcal{Q}_2$ under the constraints $\Sigma$ and write $\Sigma \models \mathcal{Q}_1 \subseteq \mathcal{Q}_2$ if, for every temporal model $\mathcal{I}$ and every time point, we have $\mathsf{ans}(\mathcal{Q}_1, \mathcal{I}(t)) \subseteq \mathsf{ans}(\mathcal{Q}_2, \mathcal{I}(t))$ whenever $\mathcal{I}, t \models \Sigma$. Note that the *query satisfiability problem*—given a query $\mathcal{Q}$ over a schema $\Sigma$, to determine whether there are $\mathcal{I}$ and $t$ such that $\mathcal{I}, t \models \Sigma$ and $\mathsf{ans}(\mathcal{Q}, \mathcal{I}(t)) \neq \emptyset$—is reducible to query containment: $\mathcal{Q}$ is satisfiable iff $\Sigma \not\models \mathcal{Q}(\overrightarrow{x}) \subseteq P(\overrightarrow{x}) \wedge \neg P(\overrightarrow{x})$, where $P$ is a $\mathcal{DLR}_{\mathcal{US}}$-relation of the same arity as $\mathcal{Q}$.

# 4 Example

As an example, let us consider the following conceptual schema $\Sigma$, where we introduce a shortcut for global atomic formulas $E_1 \sqsubseteq^* E_2 \equiv \Box^*(E_1 \sqsubseteq E_2)$, for both entities and relations:

$$
\begin{array}{l}
\texttt{Works-for} \sqsubseteq^* \texttt{emp/2} : \texttt{Employee} \sqcap \texttt{act/2} : \texttt{Project} \\
\texttt{Manages} \sqsubseteq^* \texttt{man/2} : \texttt{TopManager} \sqcap \texttt{prj/2} : \texttt{Project} \\
\texttt{Employee} \sqsubseteq^* \exists^{=1}[\texttt{worker}](\texttt{PaySlipNumber} \sqcap \texttt{num/2} : \texttt{Integer}) \sqcap \\
\qquad\qquad\quad \exists^{=1}[\texttt{payee}](\texttt{Salary} \sqcap \texttt{amount/2} : \texttt{Integer}) \\
\top \sqsubseteq^* \exists^{\leq 1}[\texttt{num}](\texttt{PaySlipNumber} \sqcap \texttt{worker/2} : \texttt{Employee}) \\
\texttt{Manager} \sqsubseteq^* \texttt{Employee} \sqcap (\texttt{AreaManager} \sqcup \texttt{TopManager}) \\
\texttt{AreaManager} \sqsubseteq^* \texttt{Manager} \sqcap \neg\texttt{TopManager} \\
\texttt{TopManager} \sqsubseteq^* \texttt{Manager} \sqcap \exists^{=1}[\texttt{man}]\texttt{Manages} \\
\texttt{Project} \sqsubseteq^* \exists^{\geq 1}[\texttt{act}]\texttt{Works-for} \sqcap \exists^{=1}[\texttt{prj}]\texttt{Manages} \\
\texttt{Employee} \sqcap \neg(\exists^{\geq 1}[\texttt{emp}]\texttt{Works-for}) \sqsubseteq^* \texttt{Manager} \\
\texttt{Manager} \sqsubseteq^* \neg(\exists^{\geq 1}[\texttt{emp}]\texttt{Works-for}) \sqcap (\texttt{Qualified}\ \mathcal{S}\ (\texttt{Employee} \sqcap \neg\texttt{Manager}))
\end{array}
$$

The theory introduces `Works-for` as a binary relation between employees and projects, and `Manages` as a binary relation between managers and projects. Employees have exactly one pay slip number and one salary each, which are represented as binary relations with an integer domain; moreover, a pay slip number uniquely identifies an employee (it acts as a key). It is stated that managers are employees, and are partitioned into area managers and top managers. Top Managers participate exactly once in the relation `Manages`, i.e., every top manager manages exactly one project. Projects participate at least once to the relation `Works-for` and exactly once in the relation `Manages`. Finally, employees not working for a project are exactly the managers, and managers should be qualified, i.e., should have passed a period of being employees.

The conceptual schema $\Sigma$ globally logically implies that, for every project, there is at least one employee who is not a manager, and that a top manager worked in a project before managing some (possibly different) project:

$$
\begin{array}{l}
\Sigma \models \texttt{Project} \sqsubseteq^* \exists^{\geq 1}[\texttt{act}](\texttt{Works-for} \sqcap \texttt{emp} : \neg\texttt{Manager}) \\
\Sigma \models \texttt{TopManager} \sqsubseteq^* \Diamond^- \exists^{\geq 1}[\texttt{emp}](\texttt{Works-for} \sqcap \texttt{act} : \texttt{Project})
\end{array}
$$

Note also that if we add to $\Sigma$ the formula

$$\texttt{Employee} \sqsubseteq^* \exists^{\geq 1}[\texttt{emp}]\texttt{Works-for}$$

saying that every employee should work for at least one project, then all the entities and the relations mentioned in the conceptual schema are interpreted as the empty set in every model of $\Sigma$, i.e., they are not satisfiable relative to $\Sigma$.

We now consider the problem of query containment under constraints, where the constraints are expressed by the above exemplified schema $\Sigma$. Consider the following queries

$$\mathcal{Q}_1(\mathtt{x},\mathtt{y})\mathtt{:-}\ \neg\mathtt{AreaManager}(\mathtt{x}) \wedge \mathtt{Manages}(\mathtt{x},\mathtt{z}) \wedge \mathtt{Project}(\mathtt{z}) \wedge \\ \mathtt{Resp\text{-}for}(\mathtt{y},\mathtt{z}) \wedge \mathtt{Department}(\mathtt{y})$$

$$\mathcal{Q}_2(\mathtt{x},\mathtt{y})\ \mathtt{:-} \\ (\Diamond^- \exists^{\geq 1}[1]\mathtt{Works\text{-}for})(\mathtt{x}) \wedge \mathtt{Manages}(\mathtt{x},\mathtt{z}) \wedge \\ \mathtt{Resp\text{-}for}(\mathtt{y},\mathtt{z}) \wedge \neg\mathtt{InterestGroup}(\mathtt{y})$$

It is not hard to see that these two queries are equivalent under the constraints in $\Sigma$, i.e.,

$$\Sigma \models \mathcal{Q}_1 \subseteq \mathcal{Q}_2 \quad \text{and} \quad \Sigma \models \mathcal{Q}_2 \subseteq \mathcal{Q}_1.$$

# 5 Decidability and complexity

In this section we only summarise the computational behaviour of $\mathcal{DLR}_{\mathcal{US}}$ and its fragments over the flow of time $\langle \mathbb{Z}, < \rangle$. Unfortunately, full $\mathcal{DLR}_{\mathcal{US}}$, even restricted to *atomic* formulas, turns out to be undecidable.

**Theorem 1.** *The global satisfiability problem for $\mathcal{DLR}_{\mathcal{US}}$ conceptual schemas containing only atomic formulas is undecidable.*

It follows, in particular, that (a) the general problem of formula satisfiability in $\mathcal{DLR}_{\mathcal{US}}$ is undecidable, and (b) the general problem of global logical implication in $\mathcal{DLR}_{\mathcal{US}}$—even involving only atomic formulas—is undecidable as well.

The main technical reason for undecidability is the possibility of temporalising binary relations. In fact, the proof uses a very small fragment of $\mathcal{DLR}_{\mathcal{US}}$: even $\mathcal{ALC}$ with $\Box^+$ or one global role is enough to get undecidability; see [25]. This gives us some grounds to conjecture that already the basic temporal EER model with just snapshot relations is undecidable.

The fragment $\mathcal{DLR}^-_{\mathcal{US}}$, in which the temporal operators can be applied only to entities and formulas, exhibits a much better computational behaviour. In this case we have the following hierarchy:

**Theorem 2.** *Let the flow of time be $\langle \mathbb{Z}, < \rangle$. Then*

(1) *the problem of logical implication in $\mathcal{DLR}^-_{\mathcal{US}}$ involving only atomic formulas is EXPTIME-complete;*

(2) *the formula satisfiability problem (and so the problem of logical implication) in $\mathcal{DLR}^-_{\mathcal{US}}$ is EXPSPACE-complete;*

(3) *the query-containment problem for non-recursive Datalog queries under $\mathcal{DLR}^-_{\mathcal{US}}$-constraints is decidable in 2EXPTIME and is EXPSPACE-hard.*

In this paper we do not go into the details of the proof; the complete proof can be found in a technical report [4]. The main technical tool in the proof is the method of quasimodels developed in [24, 26]. The idea behind the notion of a quasimodel is to represent 'the state' of the (in general, infinite) domain of a temporal model at a each moment of time by finitely many 'types' of the domain objects at this moment (modulo a given finite set of formulas); the evolution of types in time is described by special functions called runs.

# 6 Conclusion

This work introduces the temporal description logic $\mathcal{DLR}_{\mathcal{US}}$. A temporal query language was defined and the problem of query containment under the constraints defined by a $\mathcal{DLR}_{\mathcal{US}}$ conceptual schema is investigated.

# References


[1] S. Abiteboul, L. Herr, and J. Van den Bussche. Temporal versus first-order logic to query temporal databases. In *Proc. of the 15th ACM SIGACT SIGMOD SIGART Sym. on Principles of Database Systems (PODS'96)*, pages 49–57, 1996.

[2] A. Artale and E. Franconi. Temporal ER modeling with description logics. In *Proc. of the International Conference on Conceptual Modeling (ER'99)*. Springer-Verlag, November 1999.

[3] A. Artale and E. Franconi. A survey of temporal extensions of description logics. *Annals of Mathematics and Artificial Intelligence*, 30(1-4), 2001.

[4] A. Artale, E. Franconi, M. Mosurovic, F. Wolter, and M. Zakharyaschev. Reasoning over conceptual schemas and queries in temporal databases. Technical report, Dept. of Computer Science, Univ. of Manchester, UK, 2001.

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

[6] D. Calvanese, M. Lenzerini, and D. Nardi. Description logics for conceptual data modeling. In Chomicki and Saake [9].

[7] D. Calvanese, M. Lenzerini, and D. Nardi. Unifying class-based representation formalisms. *J. of Artificial Intelligence Research*, 11:199–240, 1999.

[8] J. Chomicki. Temporal query languages: a survey. In *Proc. of the 1st International Conference on Temporal Logic (ICTL'94)*, pages 506–534, 1994.

[9] J. Chomicki and G. Saake, editors. *Logics for Databases and Information Systems*. Kluwer, 1998.

[10] J. Chomicki and D. Toman. Temporal logic in information systems. In Chomicki and Saake [9], chapter 1.

[11] E. Franconi and G. Ng. The ICOM tool for intelligent conceptual modelling. In *Proc. of the $7^{th}$ International Workshop on Knowledge Representation meets Databases (KRDB'2000)*, 2000.

[12] D. M. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic: mathematical Foundations and Computational Aspects*. Oxford University Press, 1994.

[13] H Gregersen and J. S. Jensen. Temporal Entity-Relationship models - a survey. *IEEE Transactions on Knowledge and Data Engineering*, 11(3):464–497, 1999.

[14] I. Hodkinson, F. Wolter, and M. Zakharyaschev. Decidable fragments of first-order temporal logics. *Annals of Pure and Applied Logic*, 106:85–134, 2000.

[15] I. Horrocks, U. Sattler, S. Tessaris, and S. Tobies. How to decide query containment under constraints using a description logic. In *Proceedings of the 7th International Conference on Logic for Programming and Automated Reasoning (LPAR'00)*, 2000.

[16] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In *Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning (LPAR'99)*, pages 161–180, 1999.

[17] M. Jarke, C. Quix, D. Calvanese, M. Lenzerini, E. Franconi, S. Ligoudistiano, P. Vassiliadis, and Y. Vassiliou. Concept based design of data warehouses: The DWQ demonstrators. In *2000 ACM SIGMOD Intl. Conference on Management of Data*, 2000.

[18] C. S. Jensen and R. T. Snodgrass. Temporal data management. *IEEE Transactions on Knowledge and Data Engineering*, 111(1):36–44, 1999.

[19] C. S. Jensen, M. Soo, and R. T. Snodgrass. Unifying temporal data models via a conceptual model. *Information Systems*, 9(7):513–547, 1994.

[20] K. Schild. Combining terminological logics with tense logic. In *Proceedings of the $6^{th}$ Portuguese Conference on Artificial Intelligence, EPIA'93*, October 1993.

[21] A. S. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *J. Assoc. Comput. Mach.*, 32(3):733–749, 1985.

[22] S. Spaccapietra, C. Parent, and E. Zimanyi. Modeling time from a conceptual perspective. In *Int. Conf. on Information and Knowledge Management (CIKM98)*, 1998.

[23] H. Sturm and F. Wolter. A tableau calculus for temporal description logic: the expanding domain case. *Journal of Logic and Computation*, 2001. To appear.

[24] F. Wolter and M. Zakharyaschev. Satisfiability problem in description logics with modal operators. In *Proc. of the $6^{th}$ International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 512–523, Trento, Italy, June 1998.

[25] F. Wolter and M. Zakharyaschev. Modal description logics: Modalizing roles. *Fundamenta Informaticae*, 39(4):411–438, 1999.

[26] F. Wolter and M. Zakharyaschev. Multi-dimensional description logics. In *Proc. of IJCAI'99*, pages 104–109, 1999.

[27] F. Wolter and M. Zakharyaschev. Temporalizing description logics. In D. Gabbay and M. de Rijke, editors, *Frontiers of Combining Systems*. Studies Press-Wiley, 1999.