Tailoring Temporal Description Logics for Reasoning over Temporal Conceptual Models

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

[1] KRDB Research Centre
Free University of Bozen-Bolzano, Italy
{*lastname*}@inf.unibz.it

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

Abstract. Temporal data models have been used to describe how data can evolve in the context of temporal databases. Both the Extended Entity-Relationship (EER) model and the Unified Modelling Language (UML) have been temporally extended to design temporal databases. To automatically check quality properties of conceptual schemas various encoding to Description Logics (DLs) have been proposed in the literature. On the other hand, reasoning on temporally extended DLs turn out to be too complex for effective reasoning ranging from 2ExpTime up to undecidable languages. We propose here to temporalize the 'light-weight' *DL-Lite* logics obtaining nice computational results while still being able to represent various constraints of temporal conceptual models. In particular, we consider temporal extensions of $\textit{DL-Lite}^{\mathcal{N}}_{bool}$, which was shown to be adequate for capturing non-temporal conceptual models without relationship inclusion, and its fragment $\textit{DL-Lite}^{\mathcal{N}}_{core}$ with most primitive concept inclusions, which are nevertheless enough to represent almost all types of atemporal constraints (apart from covering).

1 Introduction

Conceptual data modelling formalisms such as the Unified Modelling Language (UML) and the Extended Entity-Relationship (EER) model have become a *de facto* standard in database design and software engineering by providing visual means to describe application domains in a declarative and reusable way. Both UML and EER turn out to be closely connected to description logics (DLs), which can encode constraints expressible in these conceptual modelling formalisms (see, e.g., [11, 12, 1]). This encoding provides us with a rigorous definition of various *quality properties* of conceptual schemas. For instance, given a conceptual schema, we can check its *consistency* (i.e., whether its constraints contain no contradictions), entity and relationship *satisfiability* (i.e., whether given entities and relationships in the schema can be instantiated), *instance checking* (i.e., whether a given individual belongs to a given entity in every instance of the schema), and *logical entailment* (i.e., whether a given constraint is logically implied by the schema). The encoding of conceptual models as DL knowledge bases (KBs) opens a way for utilizing existing DL reasoning services (reasoners) for automated checking of these quality properties, and so for providing an effective reasoning support for the construction phase of a conceptual model schema.

Temporal conceptual data models [31, 20, 21, 4, 26, 6, 15, 7, 10] extend standard conceptual schemas with means to visually represent temporal constraints imposed on temporal database instances. Temporal constraints can be grouped in three categories: timestamping, evolution and temporal cardinality constraints. *Timestamping constraints* discriminate between those entities, r relationships and attributes that change over time and those that are time-invariant [31, 21, 16, 7, 26]. *Evolution constraints* control how the domain elements evolve over time by 'migrating' from one entity to another [22, 25, 29, 26, 6]. We distinguish between *quantitative* evolution constraints that specify the exact time of migration and *qualitative* evolution constraints that describe eventual temporal behaviour (i.e., whether all instances will eventually migrate or will always belong to the same entity). *Temporal cardinality constraints* restrict the number of times an instance participates in a relationship; *snapshot* cardinality constraints do it at each moment of time, while *lifespan* cardinality constraints impose restrictions over the entire existence of the instance [30, 24].

Temporal conceptual models can be encoded in various temporal description logics (TDLs), which have been designed and investigated since the seminal paper [28] with the aim of understanding the computational price of introducing a temporal dimension in DLs (see [23] for a survey). A general conclusion one can draw from the obtained results is that—as far as there is a nontrivial interaction between the temporal and DL components—TDLs based on full-fledged DLs like $\mathcal{ALC}$ turn out to be too complex for effective reasoning ranging from 2EXPTIME up to undecidable languages.

The aim of this paper is to show how temporalizing the 'light-weight' *DL-Lite* logics [13, 14, 27, 2, 3] we can represent various constraints of temporal conceptual models. In particular, we consider $\textit{DL-Lite}_{bool}^{\mathcal{N}}$, which was shown to be adequate for capturing non-temporal conceptual models without relationship inclusion [1], and its fragment $\textit{DL-Lite}_{core}^{\mathcal{N}}$ with most primitive concept inclusions, which are nevertheless enough to represent almost all types of atemporal constraints (apart from covering). To capture temporal constraints, we interpret the TDLs over the flow of time $(\mathbb{Z}, <)$, in which (1) the future and past temporal operators can be applied to concepts (entities); (2) roles can be declared flexible or rigid; (3) the 'undirected' temporal operators 'always' and 'some time' can be applied to roles; (4) the concept inclusions (TBox) hold at all moments of time (i.e., global) and the database assertions (ABox) are specified to hold at particular moments of time.

Complexity results for reasoning in TDLs based on *DL-Lite* have been presented in [5, 8, 9]. The most expressive TDL based on $\textit{DL-Lite}_{bool}^{\mathcal{N}}$ and featuring all of (1)–(4) turns out to be undecidable. This 'negative' result has motivated our study of various fragments of the full language by restricting not only the DL but also the temporal component. Concerning TDLs with temporalized roles, in addition to the undecidability result, we have also shown that using the undirected temporal operators always/sometime together with temporalized roles over $\textit{DL-Lite}_{bool}^{\mathcal{N}}$ results in an NP-complete language. TDLs with rigid (and flexible but not temporalized) roles turned out to be reducible to propositional linear temporal logic $\mathcal{LTL}$ (and its natural fragments). The absence of temporalized roles makes reasoning in these logics easier with complexity results ranging from NLOGSPACE to PSPACE.

2 *DL-Lite* logics

We briefly introduce *DL-Lite* and its relatives (see [14, 3] for more details). The language of *DL-Lite*$^{\mathcal{N}}_{bool}$ contains *object names* $a_0, a_1, \dots$, *concept names* $A_0, A_1, \dots$, and *role names* $P_0, P_1, \dots$. *Roles* R, *basic concepts* B and *concepts* C of this language are defined by the rules:

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

where q is a positive integer. A *DL-Lite*$^{\mathcal{N}}_{bool}$ *TBox*, $\mathcal{T}$, is a finite set of *concept inclusion axioms* of the form

$$C_1 \sqsubseteq C_2.$$

An *ABox*, $\mathcal{A}$, is a finite set of assertions of the form

$$A_k(a_i), \qquad \neg A_k(a_i), \qquad P_k(a_i, a_j), \qquad \neg P_k(a_i, a_j).$$

Taken together, $\mathcal{T}$ and $\mathcal{A}$ constitute the *DL-Lite*$^{\mathcal{N}}_{bool}$ *knowledge base* (KB, for short) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$.

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of this and other *DL-Lite* languages consists of a *domain* $\Delta^{\mathcal{I}} \neq \emptyset$ and an interpretation function $\cdot^{\mathcal{I}}$ that assigns to each object name a_i an element $a_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, to each concept name A_k a subset $A_k^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and to each role name P_k a binary relation $P_k^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. As in databases, we adopt the *unique name assumption* (UNA) according to which $a_i^{\mathcal{I}} \neq a_j^{\mathcal{I}}$ for all $i \neq j$. The role and concept constructs are interpreted in $\mathcal{I}$ as follows:

$$\begin{array}{rcl} (P_k^-)^{\mathcal{I}} & = & \{(y,x) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (x,y) \in P_k^{\mathcal{I}}\}, \\ \bot^{\mathcal{I}} & = & \emptyset, \\ (\geq q\, R)^{\mathcal{I}} & = & \big\{x \in \Delta^{\mathcal{I}} \mid \sharp\{y \in \Delta^{\mathcal{I}} \mid (x,y) \in R^{\mathcal{I}}\} \geq q\big\}, \\ (\neg C)^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}, \\ (C_1 \sqcap C_2)^{\mathcal{I}} & = & C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}, \end{array}$$

where $\sharp X$ denotes the cardinality of X. The *satisfaction relation* $\models$ is defined as usual:

$$\begin{array}{rll} \mathcal{I} \models C_1 \sqsubseteq C_2 & \text{iff} & C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}, \\ \mathcal{I} \models A_k(a_i) & \text{iff} & a_i^{\mathcal{I}} \in A_k^{\mathcal{I}}, \qquad\qquad \mathcal{I} \models \neg A_k(a_i) \quad \text{iff} \quad a_i^{\mathcal{I}} \notin A_k^{\mathcal{I}}, \\ \mathcal{I} \models P_k(a_i, a_j) & \text{iff} & (a_i^{\mathcal{I}}, a_j^{\mathcal{I}}) \in P_k^{\mathcal{I}}, \quad \mathcal{I} \models \neg P_k(a_i, a_j) \quad \text{iff} \quad (a_i^{\mathcal{I}}, a_j^{\mathcal{I}}) \notin P_k^{\mathcal{I}}. \end{array}$$

A knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is said to be *satisfiable* (or *consistent*) if there is an interpretation, $\mathcal{I}$, satisfying all the members of $\mathcal{T}$ and $\mathcal{A}$. In this case we write $\mathcal{I} \models \mathcal{K}$ (as well as $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$) and say that $\mathcal{I}$ is a *model of* $\mathcal{K}$ (and of $\mathcal{T}$ and $\mathcal{A}$).

The two sub-languages of *DL-Lite*$^{\mathcal{N}}_{bool}$ we deal with in this article are obtained by restricting the Boolean operators on concepts. In *DL-Lite*$^{\mathcal{N}}_{krom}$ TBoxes,[1] concept inclusions are of the form

$$B_1 \sqsubseteq B_2, \qquad B_1 \sqsubseteq \neg B_2 \qquad \text{or} \qquad \neg B_1 \sqsubseteq B_2. \qquad (\textit{krom})$$

(Here and below the B_i are basic concepts.) In *DL-Lite*$^{\mathcal{N}}_{core}$, we can only use concept inclusions of the form

$$B_1 \sqsubseteq B_2 \qquad \text{or} \qquad B_1 \sqcap B_2 \sqsubseteq \bot. \qquad (\textit{core})$$

As $B_1 \sqsubseteq \neg B_2$ is equivalent to $B_1 \sqcap B_2 \sqsubseteq \bot$, *DL-Lite*$^{\mathcal{N}}_{core}$ is a sub-language of *DL-Lite*$^{\mathcal{N}}_{krom}$.

The extra expressive power, gained from covering constraints, comes at a price: the satisfiability problem is NLogSpace-complete for *DL-Lite*$^{\mathcal{N}}_{core}$ and *DL-Lite*$^{\mathcal{N}}_{krom}$ KBs and NP-complete for *DL-Lite*$^{\mathcal{N}}_{bool}$ KBs [2].

3 Temporal Conceptual Modelling

Temporal conceptual data models extend standard conceptual schemas with means to visually represent temporal constraints imposed on temporal database instances [31, 20, 21, 4, 26]. When introducing a temporal dimension into conceptual data models, time is usually modelled by a linearly ordered set of time instants, so that at each moment we can refer to its past and its future. We assume that the flow of time is isomorphic to the strictly linearly ordered set $(\mathbb{Z}, <)$ of integer numbers. (For a survey of other options, including various interval-based and branching models of time, consult, e.g. [18, 19, 17].)

A basic assumption made in temporal conceptual models is that entities, relationships and attributes may freely change over time—as long as they satisfy the schema constraints at *each* time instant. Temporal constructs are then used to impose constraints on the temporal behaviour of various components of conceptual schemas. We group these constructs into three categories—*timestamping*, *evolution constraints* and *temporal cardinality constraints*—and illustrate them using the temporal data model in Figure 1.

Timestamping constraints [31, 21, 26] distinguish between entities, relationships and attributes that are *temporary*, i.e., cannot keep a single element over the whole timeline; *snapshot*, or time-invariant; and *unconstrained* (all others). In temporal entity-relationship (TER) diagrams, temporary entities, relationships and attributes are marked with T and snapshot ones with S. In Figure 1, 'Employee' and 'Department' are snapshot entities, 'Name,' 'PaySlipNumber' and 'ProjectCode' are snapshot attributes and 'Member' a snapshot relationship. On the other hand, 'Manager' is a temporary entity, 'Salary' a temporary attribute and 'WorksOn' a temporary relationship.

[1] The Krom fragment of first-order logic consists of all formulas in prenex normal form whose quantifier-free part is a conjunction of binary clauses.

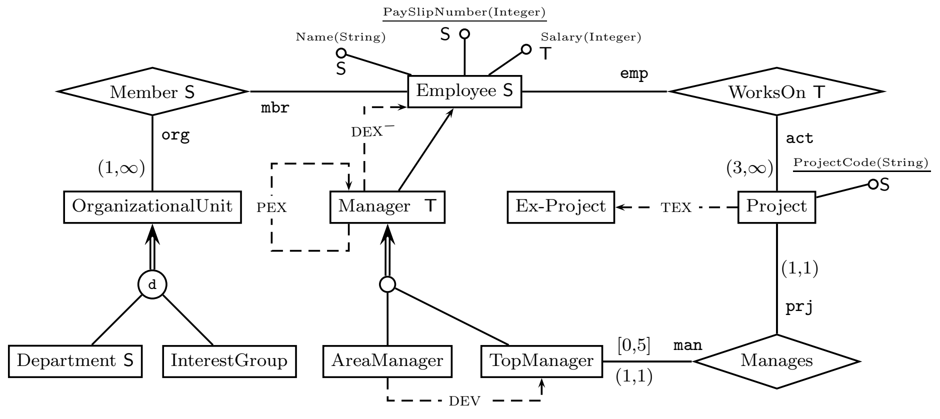

Fig. 1. A temporal conceptual model of a company information system.

To represent timestamping constraints in temporal description logics we employ the *temporal operator* $\boxast$, which is read as 'always' or 'at all—past, present and future—time instants.' Intuitively, for a concept C, $\boxast C$ contains those elements that belong to C at all time instants. Using this operator, the constraints 'Employee is a snapshot entity' and 'Manager is a temporary entity' can be represented as follows:

$$\mathit{Employee} \sqsubseteq \boxast \mathit{Employee}, \tag{1}$$

$$\boxast \mathit{Manager} \sqsubseteq \bot. \tag{2}$$

The first inclusion says that, at any moment of time, every element of 'Employee' has always been and will always be an element of 'Employee.' The second one states that no element can belong to 'Manager' at all time instants. Note that we consider concept inclusions to hold *globally*, that is, at all moments of time.

The same temporal operator $\boxast$ together with rigid roles (i.e., roles that do not change over time) can be used to capture timestamping of (reified) relationships. Rigid roles can also represent snapshot attributes, while temporary attributes can be captured by using temporalized roles: $\exists \boxast \mathit{salary} \sqsubseteq \bot$, where $\boxast \mathit{salary}$ denotes the intersection of the relations *salary* at all time instants, model *salary* as a temporary attribute.

Evolution constraints control how the domain elements evolve over time by 'migrating' from one entity to another [22, 25, 29, 26, 6]. We distinguish between *qualitative* evolution constraints that describe eventual temporal behaviour and do not specify the moment of migration, and *quantitative* evolution (or transition) constraints that specify the exact moment of migration. The dashed arrow marked with TEX in Figure 1 is an example of a quantitative evolution constraint meaning that each 'Project' expires in exactly one year and becomes an 'Ex-Project.' The dashed arrow marked with DEV is a qualitative evolution constraint meaning that every 'AreaManager' will eventually (at some moment in the future) become a 'TopManager.' The DEX$^-$ dashed arrow says that every 'Manager' was once an 'Employee,' while the PEX dashed arrow means that a 'Manager' will always be a 'Manager' and cannot be demoted. In temporal description logic, these evolution constraints are represented using temporal operators such as 'at the next moment of time' $\bigcirc_F$, 'some time in the future' $\Diamond_F$, 'some time in the past' $\Diamond_P$ and 'always in the future' $\Box_F$:

$$\begin{aligned} \textit{Project} &\sqsubseteq \bigcirc_F \textit{Ex-Project}, && (3)\\ \textit{AreaManager} &\sqsubseteq \Diamond_F \textit{TopManager}, && (4)\\ \textit{Manager} &\sqsubseteq \Diamond_P \textit{Employee}, && (5)\\ \textit{Manager} &\sqsubseteq \Box_F \textit{Manager}. && (6) \end{aligned}$$

We note again that these concept inclusions hold at every moment of time.

Temporal cardinality constraints [30, 24, 20] restrict the number of times an instance participates in a relationship. *Snapshot* cardinality constraints do it at each moment of time, while *lifespan* cardinality constraints impose restrictions over the entire existence of the instance. In Figure 1, we use (k, l) to specify the snapshot cardinalities and $[k, l]$ the lifespan cardinalities: for example, every 'TopManager' manages exactly one project at each moment of time (snapshot cardinality), but not more than five different projects over the whole career (lifespan cardinality). If the relationship 'manages' is represented by a role in temporal description logic then these two constraints can be expressed by the following concept inclusions:

$$\begin{aligned} \textit{TopManager} &\sqsubseteq\ \leq 1\, \textit{manages},\\ \textit{TopManager} &\sqsubseteq\ \leq 5\, \Diamond\!\!\!\!\Diamond \textit{manages}, \end{aligned}$$

where $\Diamond\!\!\!\!\Diamond$ means 'sometime' (in the past, present or future), and so $\Diamond\!\!\!\!\Diamond$*manages* is the union of the relations *manages* over *all* time instants.

Finally, to represent temporal database instances associated to a temporal conceptual model, we use assertions like $\bigcirc_P \textit{Manager}(\textit{bob})$ for 'Bob was a manager last year' and $\bigcirc_F \textit{manages}(\textit{bob}, \textit{cronos})$ for 'Bob will manage project Cronos next year.'

3.1 Temporal *DL-Lite* logics

It is known from temporal logic [18] that all the temporal operators used in the previous section can be expressed in terms of the binary operators 'since' $\mathcal{S}$ and 'until' $\mathcal{U}$. So we formulate our 'base' temporal extension $T_{\mathcal{US}}\textit{DL-Lite}^{\mathcal{N}}_{\textit{bool}}$ of the description logic $\textit{DL-Lite}^{\mathcal{N}}_{\textit{bool}}$ using only these two operators. The language of $T_{\mathcal{US}}\textit{DL-Lite}^{\mathcal{N}}_{\textit{bool}}$ contains *object names* $a_0, a_1, \dots$, *concept names* $A_0, A_1, \dots$, *flexible role names* $P_0, P_1, \dots$ and *rigid role names* $G_0, G_1, \dots$. *Role names* S, *roles* R, *basic concepts* B, *concepts* C and *temporal concepts* D are defined by the following rules:

$$\begin{aligned} S &::= P_i \mid G_i,\\ R &::= S \mid S^-,\\ B &::= \bot \mid A_i \mid \geq q\, R,\\ C &::= B \mid D \mid \neg C \mid C_1 \sqcap C_2,\\ D &::= C \mid C_1 \,\mathcal{U}\, C_2 \mid C_1 \,\mathcal{S}\, C_2, \end{aligned}$$

where, as before, q is a positive integer. A *$T_{\mathcal{US}}$DL-Lite$^{\mathcal{N}}_{bool}$ TBox*, $\mathcal{T}$, is a finite set of *concept inclusions* of the form $C_1 \sqsubseteq C_2$. An *ABox*, $\mathcal{A}$, consists of assertions of the form

$$\bigcirc^n A_k(a_i), \qquad \bigcirc^n \neg A_k(a_i), \qquad \bigcirc^n S(a_i, a_j) \quad \text{and} \quad \bigcirc^n \neg S(a_i, a_j),$$

where A_k is a concept name, S a (flexible or rigid) role name, a_i, a_j object names and, for $n \in \mathbb{Z}$,

$$\bigcirc^n = \underbrace{\bigcirc_F \cdots \bigcirc_F}_{n \text{ times}}, \text{ if } n \geq 0 \qquad \text{and} \qquad \bigcirc^n = \underbrace{\bigcirc_P \cdots \bigcirc_P}_{-n \text{ times}}, \text{ if } n < 0.$$

Taken together, the TBox $\mathcal{T}$ and ABox $\mathcal{A}$ form the *knowledge base* (KB) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$.

A *temporal interpretation*, $\mathcal{I}$, gives a standard DL interpretation, $\mathcal{I}(n)$, for each time instant $n \in \mathbb{Z}$:

$$\mathcal{I}(n) \;=\; \big(\Delta^{\mathcal{I}}, a_0^{\mathcal{I}}, \dots, A_0^{\mathcal{I}(n)}, \dots, P_0^{\mathcal{I}(n)}, \dots, G_0^{\mathcal{I}}, \dots\big).$$

We assume, however, that the domain $\Delta^{\mathcal{I}}$ and the interpretations $a_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ of the object names and $G_0^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ of rigid role names are fixed for all time. (Recall also that we adopt the UNA.) The interpretations $A_i^{\mathcal{I}(n)} \subseteq \Delta^{\mathcal{I}}$ of concept names and $P_i^{\mathcal{I}(n)} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ of flexible role names can vary. The atemporal constructs are interpreted in $\mathcal{I}(n)$ as before; we write $C^{\mathcal{I}(n)}$ for the extension of concept C in the interpretation $\mathcal{I}(n)$. The interpretation of the temporal operators is as in temporal logic:

$$\begin{aligned}
(C_1 \,\mathcal{U}\, C_2)^{\mathcal{I}(n)} \;&=\; \bigcup_{k>n} \big(C_2^{\mathcal{I}(k)} \cap \bigcap_{n<m<k} C_1^{\mathcal{I}(m)}\big),\\
(C_1 \,\mathcal{S}\, C_2)^{\mathcal{I}(n)} \;&=\; \bigcup_{k<n} \big(C_2^{\mathcal{I}(k)} \cap \bigcap_{n>m>k} C_1^{\mathcal{I}(m)}\big).
\end{aligned}$$

Concept inclusions are interpreted in $\mathcal{I}$ *globally*:

$$\mathcal{I} \models C_1 \sqsubseteq C_2 \quad \text{iff} \quad C_1^{\mathcal{I}(n)} \subseteq C_2^{\mathcal{I}(n)} \quad \text{for } \textit{all } n \in \mathbb{Z}.$$

And for the ABox assertions, we set:

$$\begin{aligned}
\mathcal{I} \models \bigcirc^n A_k(a_i) \ \text{ iff } \ a_i^{\mathcal{I}} \in A_k^{\mathcal{I}(n)}, \qquad & \mathcal{I} \models \bigcirc^n \neg A_k(a_i) \ \text{ iff } \ a_i^{\mathcal{I}} \notin A_k^{\mathcal{I}(n)},\\
\mathcal{I} \models \bigcirc^n S(a_i, a_j) \ \text{ iff } \ (a_i^{\mathcal{I}}, a_j^{\mathcal{I}}) \in S^{\mathcal{I}(n)}, \quad & \mathcal{I} \models \bigcirc^n \neg S(a_i, a_j) \ \text{ iff } \ (a_i^{\mathcal{I}}, a_j^{\mathcal{I}}) \notin S^{\mathcal{I}(n)}.
\end{aligned}$$

We call $\mathcal{I}$ a *model* of a KB $\mathcal{K}$ and write $\mathcal{I} \models \mathcal{K}$ if $\mathcal{I}$ satisfies all elements of $\mathcal{K}$. If $\mathcal{K}$ has a model then it is said to be *satisfiable*. A concept C (role R) is *satisfiable* w.r.t. $\mathcal{K}$ if there are a model $\mathcal{I}$ of $\mathcal{K}$ and $n \in \mathbb{Z}$ such that $C^{\mathcal{I}(n)} \neq \emptyset$ (respectively, $R^{\mathcal{I}(n)} \neq \emptyset$). It is readily seen that the concept and role satisfiability problems are equivalent to KB satisfiability.

We now define a few fragments and extensions of the base language $T_{\mathcal{US}}$*DL-Lite*$^{\mathcal{N}}_{bool}$. Recall that to say that C is a snapshot concept, we need the 'always' operator $\boxed{*}$ with the following meaning:

$$(\boxed{*}\, C)^{\mathcal{I}(n)} \;=\; \bigcap_{k \in \mathbb{Z}} C^{\mathcal{I}(k)}.$$

In terms of $\mathcal{S}$ and $\mathcal{U}$, this operator can be represented as $\boxast C = \neg(\top\ \mathcal{S}\ \neg C) \sqcap C \sqcap \neg(\top\ \mathcal{U}\ \neg C)$. Define T_U*DL-Lite*$^{\mathcal{N}}_{bool}$ to be the sublanguage of $T_{\mathcal{US}}$*DL-Lite*$^{\mathcal{N}}_{bool}$ the temporal concepts D in which are of the form:

$$D \ ::= \ C \ \mid \ \boxast C. \tag{U}$$

Thus, in T_U*DL-Lite*$^{\mathcal{N}}_{bool}$, we can express timestamping constraints (see Section 3).

The temporal operators $\Diamond_F$ ('some time in the future') and $\Diamond_P$ ('some time in the past') that are required for qualitative evolution constraints with the standard temporal logic semantics

$$(\Diamond_F C)^{\mathcal{I}(n)} \ = \ \bigcup_{k>n} C^{\mathcal{I}(k)} \quad \text{and} \quad (\Diamond_P C)^{\mathcal{I}(n)} \ = \ \bigcup_{k<n} C^{\mathcal{I}(k)}$$

can be expressed via $\mathcal{U}$ and $\mathcal{S}$ as $\Diamond_F C = \top\ \mathcal{U}\ C$ and $\Diamond_P C = \top\ \mathcal{S}\ C$; the operators $\Box_F$ ('always in the future') and $\Box_P$ ('always in the past') are defined as dual to $\Diamond_F$ and $\Diamond_P$: $\Box_F C = \neg\Diamond_F\neg C$ and $\Box_P C = \neg\Diamond_P\neg C$. We define the fragment T_{FP}*DL-Lite*$^{\mathcal{N}}_{bool}$ of $T_{\mathcal{US}}$*DL-Lite*$^{\mathcal{N}}_{bool}$ by restricting the temporal concepts D to the form:

$$D \ ::= \ C \ \mid \ \Box_F C \ \mid \ \Box_P C. \tag{FP}$$

Clearly, we have the following equivalences:

$$\boxast C = \Box_F\Box_P C \qquad \text{and} \qquad \diamond\!\!\!\!\ast\, C = \Diamond_F\Diamond_P C.$$

In what follows they will be regarded as definitions for $\boxast$ and $\diamond\!\!\!\!\ast$ in the languages, where they are not explicitly present. Thus, T_{FP}*DL-Lite*$^{\mathcal{N}}_{bool}$ is capable of expressing both timestamping and qualitative (but not quantitative) evolution constraints.

The temporal operators $\bigcirc_F$ ('next time') and $\bigcirc_P$ ('previous time'), used in quantitative evolution constraints, can be defined as $\bigcirc_F C = \bot\ \mathcal{U}\ C$ and $\bigcirc_P C = \bot\ \mathcal{S}\ C$, so that we have:

$$(\bigcirc_F C)^{\mathcal{I}(n)} = C^{\mathcal{I}(n+1)} \quad \text{and} \quad (\bigcirc_P C)^{\mathcal{I}(n)} = C^{\mathcal{I}(n-1)}.$$

The fragment of $T_{\mathcal{US}}$*DL-Lite*$^{\mathcal{N}}_{bool}$ with temporal concepts of the form

$$D \ ::= \ C \ \mid \ \Box_F C \ \mid \ \Box_P C \ \mid \ \bigcirc_F C \ \mid \ \bigcirc_P C \tag{FPX}$$

will be denoted by T_{FPX}*DL-Lite*$^{\mathcal{N}}_{bool}$. In this fragment, we can express timestamping, qualitative and quantitative evolution constraints.

We have the following inclusions between the languages:

$$T_U\textit{DL-Lite}^{\mathcal{N}}_{bool} \ \subseteq \ T_{FP}\textit{DL-Lite}^{\mathcal{N}}_{bool} \ \subseteq \ T_{FPX}\textit{DL-Lite}^{\mathcal{N}}_{bool} \ \subseteq \ T_{\mathcal{US}}\textit{DL-Lite}^{\mathcal{N}}_{bool}.$$

Similarly to the non-temporal case, we can also identify sub-Boolean fragments of the above languages. A temporal TBox $\mathcal{T}$ will be called a *Krom* (*core*) TBox if it contains only concept inclusions of the form:

$$D_1 \sqsubseteq D_2, \qquad D_1 \sqsubseteq \neg D_2, \qquad \neg D_1 \sqsubseteq D_2, \tag{Krom}$$

$$D_1 \sqsubseteq D_2, \qquad D_1 \sqcap D_2 \sqsubseteq \bot, \tag{core}$$

respectively, where the D_i are temporal concepts defined by (FPX), (FP) or (U) with $C ::= B \mid D$ (so, no Boolean operators are allowed in the D_i). This gives us 6 different fragments $T_{\mathit{FPX}}\mathit{DL\text{-}Lite}_{\alpha}^{\mathcal{N}}$, $T_{\mathit{FP}}\mathit{DL\text{-}Lite}_{\alpha}^{\mathcal{N}}$ and $T_{U}\mathit{DL\text{-}Lite}_{\alpha}^{\mathcal{N}}$, for $\alpha \in \{\mathit{core}, \mathit{krom}\}$. We do not consider the core and Krom fragments of the full language with $\mathcal{U}/\mathcal{S}$ because these operators allow one to go beyond the language of binary clauses of the core and Krom fragments; the resulting languages would have the same complexity as $T_{\mathcal{US}}\mathit{DL\text{-}Lite}_{\mathit{bool}}^{\mathcal{N}}$ and yet be less expressive (see [9] for more details).

We note here that both Krom and Bool TBoxes have the full negation, and so one can freely use $\Diamond$-shaped counterparts of the $\Box$ temporal operators allowed in the language. This is not the case for the core fragments where timestamping can still be expressed (cf. (1) and (2)) but evolution constraints involving $\Diamond$ (e.g., a *Manager* was once an *Employee*; cf. (5)) are not expressible.

Table 1. The temporal extended *DL-Lite* family and complexity of its members.

concept inclusions	temporal constructs		
	$\mathcal{U}/\mathcal{S}, \bigcirc_F/\bigcirc_P, \Box_F/\Box_P$ [a]	$\Box_F/\Box_P$	$\boxast$
Bool	$T_{\mathcal{US}}\mathit{DL\text{-}Lite}_{\mathit{bool}}^{\mathcal{N}}$ $T_{\mathit{FPX}}\mathit{DL\text{-}Lite}_{\mathit{bool}}^{\mathcal{N}}$ PSPACE	$T_{\mathit{FP}}\mathit{DL\text{-}Lite}_{\mathit{bool}}^{\mathcal{N}}$ NP	$T_{U}\mathit{DL\text{-}Lite}_{\mathit{bool}}^{\mathcal{N}}$ NP
Krom	$T_{\mathit{FPX}}\mathit{DL\text{-}Lite}_{\mathit{krom}}^{\mathcal{N}}$ NP	$T_{\mathit{FP}}\mathit{DL\text{-}Lite}_{\mathit{krom}}^{\mathcal{N}}$ NP	$T_{U}\mathit{DL\text{-}Lite}_{\mathit{krom}}^{\mathcal{N}}$ NLOGSPACE
core	$T_{\mathit{FPX}}\mathit{DL\text{-}Lite}_{\mathit{core}}^{\mathcal{N}}$ in PTIME	$T_{\mathit{FP}}\mathit{DL\text{-}Lite}_{\mathit{core}}^{\mathcal{N}}$ in PTIME	$T_{U}\mathit{DL\text{-}Lite}_{\mathit{core}}^{\mathcal{N}}$ NLOGSPACE
temporalized roles	$T_{X}^{\mathcal{R}}\mathit{DL\text{-}Lite}_{\mathit{bool}}^{\mathcal{N}}$ undec.	?	$T_{U}^{\mathcal{R}}\mathit{DL\text{-}Lite}_{\mathit{bool}}^{\mathcal{N}}$ NP

[a] Sub-boolean fragments of the language with $\mathcal{U}/\mathcal{S}$ are not defined.

As we have seen in our running example, in order to express lifespan cardinality constraints, temporal operators on roles are required: for a role R of the form

$$R \ ::= \ S \ \mid \ S^{-} \ \mid \ \diamondast R \ \mid \ \boxast R,$$

the extensions of $\diamondast R$ and $\boxast R$ in an interpretation $\mathcal{I}$ are defined as

$$(\diamondast R)^{\mathcal{I}(n)} = \bigcup_{k \in \mathbb{Z}} R^{\mathcal{I}(k)} \quad \text{and} \quad (\boxast R)^{\mathcal{I}(n)} = \bigcap_{k \in \mathbb{Z}} R^{\mathcal{I}(k)}.$$

We denote by $T_{\beta}^{\mathcal{R}}\mathit{DL\text{-}Lite}_{\mathit{bool}}^{\mathcal{N}}$, for $\beta \in \{\mathit{FPX}, \mathit{FP}, U\}$, the extensions of the respective Bool fragments with temporalized roles.

To summarize, the temporal extensions of the *DL-Lite* logics we consider in this paper are collected in Table 1. The tight (unless specified otherwise) complexity bounds of Table 1 have been established in [9].

4 Conclusions

From the complexity-theoretic point of view, the best candidates for reasoning about TCMs appear to be the TDLs $T_{FPX}\,\textit{DL-Lite}^{\mathcal{N}}_{core}$ and $T_{FPX}\,\textit{DL-Lite}^{\mathcal{N}}_{bool}$, the former of which is NP-complete and the latter PSPACE-compete. Moreover, as showed in [9], the reduction of $T_{FPX}\,\textit{DL-Lite}^{\mathcal{N}}_{core}$ to $\mathcal{LTL}$ can be done deterministically, thus standard $\mathcal{LTL}$ provers can be used for TCM reasoning. We also believe that $T_{FPX}\,\textit{DL-Lite}^{\mathcal{N}}_{core}$ extended with temporalized roles can be decidable, which remains one of the most challenging open problems. But it seems to be next to impossible to reason in an effective way about all TCM constrains without any restrictions.

References

1. A. Artale, D. Calvanese, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. Reasoning over extended ER models. In *Proc. of the* 26^{th} *Int. Conf. on Conceptual Modeling (ER'07)*, volume 4801 of Lecture Notes in Computer Science, pages 277–292. Springer, 2007.
2. A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. *DL-Lite* in the light of first-order logic. In *Proc. of the 22*nd *Nat. Conf. on Artificial Intelligence (AAAI 2007)*, pages 361–366, 2007.
3. A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. The *DL-Lite* family and relations. *J. of Artificial Intelligence Research*, 36:1–69, 2009.
4. A. Artale, E. Franconi, and F. Mandreoli. Description logics for modelling dynamic information. In J. Chomicki, R. van der Meyden, and G. Saake, editors, *Logics for Emerging Applications of Databases*. Lecture Notes in Computer Science, Springer, 2003.
5. A. Artale, R. Kontchakov, C. Lutz, F. Wolter, and M. Zakharyaschev. Temporalising tractable description logics. In *14th Int. Symposium on Temporal Representation and Reasoning (TIME07)*. IEEE Computer Society, 2007.
6. A. Artale, C. Parent, and S. Spaccapietra. Evolving objects in temporal information systems. *Annals of Mathematics and Artificial Intelligence*, 50(1-2):5–38, 2007.
7. A. Artale and E. Franconi. Foundations of temporal conceptual data models. In *Conceptual Modeling: Foundations and Applications*, volume 5600 of Lecture Notes in Computer Science. Springer, 2009.
8. A. Artale, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. *DL-Lite* with temporalised concepts, rigid axioms and roles. In *Proc. of FroCoS-09*, volume 5749 of Lecture Notes in Artificial Intelligence, pages 133–148. Springer, 2009.
9. A. Artale, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. TDL-Lite: How to cook decidable temporal description logics. *To be submitted*, 2011.
10. A. Artale, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. Complexity of reasoning over temporal data models. In *Proc. of the* 29^{th} *Int. Conf. on Conceptual Modeling (ER'10)*, volume 6412 of Lecture Notes in Computer Science, pages 174–187. Springer, 2010.
11. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003. (2nd edition, 2007).
12. D. Berardi, D. Calvanese, and G. De Giacomo. Reasoning on UML class diagrams. *Artificial Intelligence*, 168(1–2):70–118, 2005.
13. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. *DL-Lite*: Tractable description logics for ontologies. In *Proc. of the 20*th *Nat. Conf. on Artificial Intelligence (AAAI 2005)*, pages 602–607, 2005.

14. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. of Artificial Intelligence Research (JAIR)*, 39(3):385–429, 2007.
15. C. Combi, S. Degani, and C. S. Jensen. Capturing temporal constraints in temporal ER models. In *Proc. of the* 27^{th} *Int. Conf. on Conceptual Modeling (ER'08)*. Lecture Notes in Computer Science. Springer, 2008.
16. M. Finger and P. McBrien. Temporal conceptual-level databases. In D. Gabbay, M. Reynolds, and M. Finger, editors, *Temporal Logics – Mathematical Foundations and Computational Aspects*, pages 409–435. Oxford University Press, 2000.
17. D. Gabbay, A.Kurucz, F. Wolter, and M. Zakharyaschev. *Many-dimensional modal logics: theory and applications*. Studies in Logic. Elsevier, 2003.
18. D. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects, Volume 1*. Oxford University Press, 1994.
19. D. Gabbay, M. Finger, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects*, volume 2. Oxford University Press, 2000.
20. H. Gregersen and J.S. Jensen. Conceptual modeling of time-varying information. Technical Report TimeCenter TR-35, Aalborg University, Denmark, 1998.
21. H. Gregersen and J.S. Jensen. Temporal Entity-Relationship models – a survey. *IEEE Transactions on Knowledge and Data Engineering*, 11(3):464–497, 1999.
22. G. Hall and R. Gupta. Modeling transition. In *Proc. of ICDE'91*, pages 540–549, 1991.
23. C. Lutz, F. Wolter, and M. Zakharyaschev. Temporal description logics: A survey. In *Proc. of 15th Int. Symposium on Temporal Representation and Reasoning (TIME08)*. IEEE Computer Society, 2008.
24. P. McBrien, A.H. Seltveit, and B. Wangler. An Entity-Relationship model extended to describe historical information. In *Proc. of CISMOD'92*, pages 244–260, Bangalore, India, 1992.
25. A.O. Mendelzon, T. Milo, and E. Waller. Object migration. In *Proc. of the 13th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS94)*, pages 232–242, New York, NY, USA, 1994. ACM Press.
26. C. Parent, S. Spaccapietra, and E. Zimanyi. *Conceptual Modeling for Traditional and Spatio-Temporal Applications—The MADS Approach*. Springer, 2006.
27. A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. on Data Semantics*, X:133–173, 2008.
28. K. Schild. Combining terminological logics with tense logic. In *Proc. of the 6th Portuguese Conf. on Artificial Intelligence*, pages 105–120, London, UK, 1993. Springer.
29. J. Su. Dynamic constraints and object migration. *Theoretical Computer Science*, 184(1–2):195–236, 1997.
30. B. Tauzovich. Towards temporal extensions to the entity-relationship model. In *Proc. of the Int. Conf. on Conceptual Modeling (ER'91)*. Lecture Notes in Computer Science. Springer, 1991.
31. C. Theodoulidis, P. Loucopoulos, and B. Wangler. A conceptual modelling formalism for temporal database applications. *Information Systems*, 16(3):401–416, 1991.