Reasoning over Conceptual Schemas and Queries in Temporal Databases

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

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

May 2, 2002

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*. The expressive power of the resulting $\mathcal{DLR}_{\mathcal{US}}$ logic is illustrated by providing a systematic formalisation of the most important temporal entity-relationship data models appeared in the literature. 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

Temporal databases are databases that store historical information, i.e., past, present, and potential future data [Jensen and Snodgrass, 1999]. Many formalisations have been proposed for temporal databases which are based on first-order temporal logic[Chomicki and Toman, 1998]. Although these formalisations can be very useful for characterising semantical problems arising in temporal databases, say, conceptual modelling or querying, usually they are computationally unfeasible for performing deduction tasks (for example, logical implication in the first-order temporal logic of the flow of time $\langle \mathbb{Z}, < \rangle$ or $\langle \mathbb{N}, < \rangle$ is not even recursively enumerable). An obvious solution to this problem would be to look for well-behaved fragments of first-order temporal logic (see e.g.[Chomicki and Toman, 1998] and references therein); however this way has not been successful.[1] Another idea is to deviate from the first-order paradigm and start from computationally more friendly languages such as description logics which have

[1] The only promising approach we know of is the recent paper [Hodkinson *et al.*, 2000].

been used in the area of non-temporal databases to characterise in a uniform framework both conceptual modelling and queries [Levy and Rousset, 1998; Calvanese *et al.*, 1998a].

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 [Calvanese *et al.*, 1998c; 1999]. 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 [Calvanese *et al.*, 1998a; 2000]. Logical implication in $\mathcal{DLR}$ is EXPTIME-complete [Calvanese *et al.*, 1998a]; practical correct and complete algorithms exist in implemented systems which are used in real applications [Horrocks *et al.*, 1999; Horrocks, 1999; Jarke *et al.*, 2000; Franconi and Ng, 2000].

$\mathcal{DLR}$ 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 [Calvanese *et al.*, 1998a]. View-based query processing turns out to be very useful for information integration, data warehousing, query optimisation, incomplete information management [Calvanese *et al.*, 1998b]. View-based query answering requires to answer a query over a virtual database (constrained by a $\mathcal{DLR}$ theory playing the role of the conceptual schema and of the integrity constraints) for which the only information comes from a set of materialised views over the same database. Query answering with non-recursive Datalog queries and views referring to predicates defined in a $\mathcal{DLR}$ theory is a co-NP-complete problem (in data complexity) under the closed world assumption. Checking query containment of non-recursive Datalog queries is decidable in 2EXPTIME [Calvanese *et al.*, 1998a; Horrocks *et al.*, 2000].

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* [Sistla and Clarke, 1985; Gabbay *et al.*, 1994] (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) [Schild, 1993; Wolter and Zakharyaschev, 1998; 1999c]. For the $\mathcal{ALC}$ fragment a tableau based algorithm has been studied in [Sturm and Wolter, 2001]. For a survey of various approaches to temporal description logics see [Artale and Franconi, 2001].

To illustrate the expressive power of $\mathcal{DLR}_{\mathcal{US}}$, we provide 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 [Gregersen and Jensen, 1998; 1999; Spaccapietra *et al.*, 1998; Theodoulidis *et al.*, 1991; Artale and Franconi, 1999b]. 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.

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

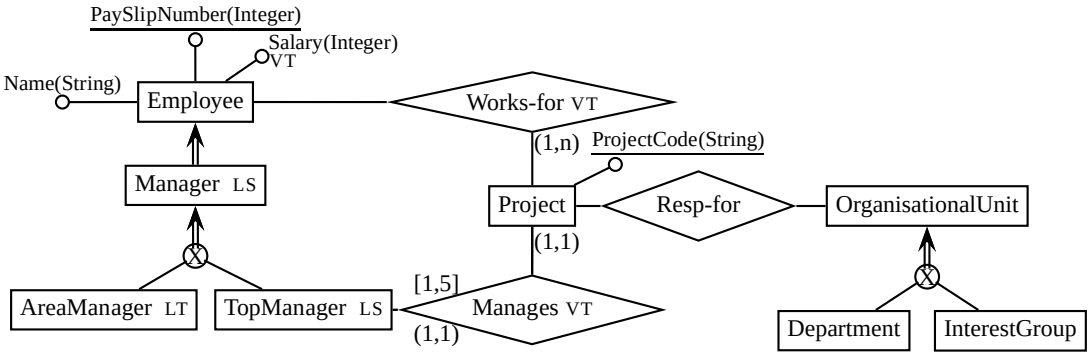

Figure 1: A TIMEER diagram.

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. [Wolter and Zakharyaschev, 1999a; Hodkinson *et al.*, 2000]). 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 in this paper, 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 paper is organised as follows. . . .

2 Modelling Temporal Databases

Temporally enhanced ER models have been developed to conceptualise the temporal aspects of database schemas, namely *valid time* – when a fact holds, i.e., it is true in the representation of the world – and *transaction time* – which records the history of database states rather than the world history, i.e., is the time when a fact is *current* in the database and can be retrieved.

In the temporal ER community two different main modelling approaches have been devised to provide temporal support. The *implicit* approach hides the temporal dimension in the interpretation structure of the ER constructs. Thus, a temporal ER model does not include any new specific temporal construct with respect to a standard non-temporal ER model. Each ER construct is always interpreted with a temporal semantics, so that instances of entities or relationships are always potentially time-varying objects.

The *explicit* approach, on the other hand, retains the non-temporal semantics for the conventional ER constructs, while adding new syntactical constructs for representing temporal entities and relationships and their temporal interdependencies. The advantage of the explicit approach is the so called *upward compatibility*: the meaning of conventional (legacy) ER diagrams when used inside a temporal model remains unchanged. This crucial property is not realizable within a strict implicit temporal approach. Indeed, if the implicit approach has the attractive of leaving unchanged the original ER model, "In its extreme, this approach rules out the possibility of designing non-temporal databases or databases where some part of a database is non-temporal and the rest is temporal"[Gregersen and Jensen, 1999].

Among the different temporal ER models presented in the literature we give here a brief overview of three of them: TIMEER [Gregersen and Jensen, 1998], ERT [Theodoulidis *et al.*, 1991], MADS [Spaccapietra *et al.*, 1998]. These models have been chosen both because they are closely related to our proposal, and they cover the full spectrum of temporal constructs (cf.[Gregersen and Jensen, 1999] for an extensive overview of temporally extended ER models).

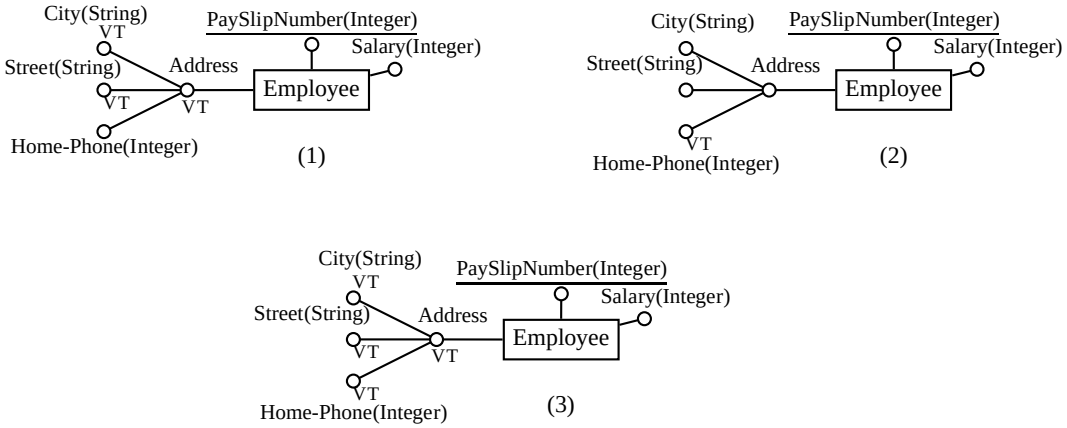

Figure 2: Temporal composite attributes in MADS.

TIMEER [Gregersen and Jensen, 1998] supports both valid and transaction time for entities, relationships and attributes. It can be regarded as an extension of ERT [Theodoulidis *et al.*, 1991] where only valid time is considered, and only binary relationships are allowed. Both TIMEER and ERT introduce new temporal constructs in the *explicit* philosophy to support the temporal dimension. While ERT annotates temporal schema elements with the label T, TIMEER uses the following labels: LS for lifespan—i.e., the validity time for entities; VT for valid time of relationships and attributes; TT for transaction time of entities, relationships and attributes; LT for both lifespan and transaction time of entities; BT for both valid and transaction time of relationships and attributes. Cardinality constraints are distinguished into *snapshot* and *lifespan* cardinality constraints [Tauzovich, 1991]. Snapshot cardinalities—represented by (min, max)—define the minimum and maximum number of participation of an entity in a relationship at *each* point in time, while lifespan cardinalities—represented by $[min, max]$—are evaluated along the entire existence time of the entity. Figure 1 gives a TIMEER diagram showing the various temporal constructs, and will form a running example trough the paper. `Manager` is a sub-entity of `Employee`—represented with a line with an arrow pointing to the super-entity—and is partitioned into `AreaManager` and `TopManager`—disjoint covering hierarchies are represented with a crossed circle (disjoint) together with a double ISA line (covering). Both lifespan and transaction time are captured for `AreaManager`, while just lifespan is considered for both `Manager` and `TopManager`. No temporal aspects are considered for `Employee` which has two integer attributes: `PaySlipNumber`, that is the key, and `Salary` for which validity time is captured. `Works-for` and `Manages` are two binary relationships for which validity time is captured. The participation of `TopManager` in the `Manages` relationship is constrained by both snapshot $(1, 1)$ and lifespan $[1, 5]$ cardinality constraints stating that top managers should manage at most 5 different projects in their entire existence while still being constrained in managing exactly one project at a time. `Resp-for` is an atemporal relationship between two atemporal entities: `Project` and `OrganisationalUnit`.

MADS [Spaccapietra *et al.*, 1998] supports valid time: entities, attributes and relationships can be annotated with a clock symbol to indicate that their extensions have a life cycle (to adopt a uniform notation we substitute the clock with the VT mark). Differently from TIMEER, components of a composite attribute can have different temporal behaviours. This is in accordance with the *identity principle*[2] for objects whose change of a property/component doesn't necessarily change the identity of the object itself. For example, for the composite attribute `Address`, composed of `City`, `Street`, `Home-Phone`

[2]See [Simons, 1987] for a comprehensive overview of the relevant philosophical literature in this area. This property is a particular case of the *orthogonality principle* as introduced in MADS advocating the independence between the different temporal constructors.

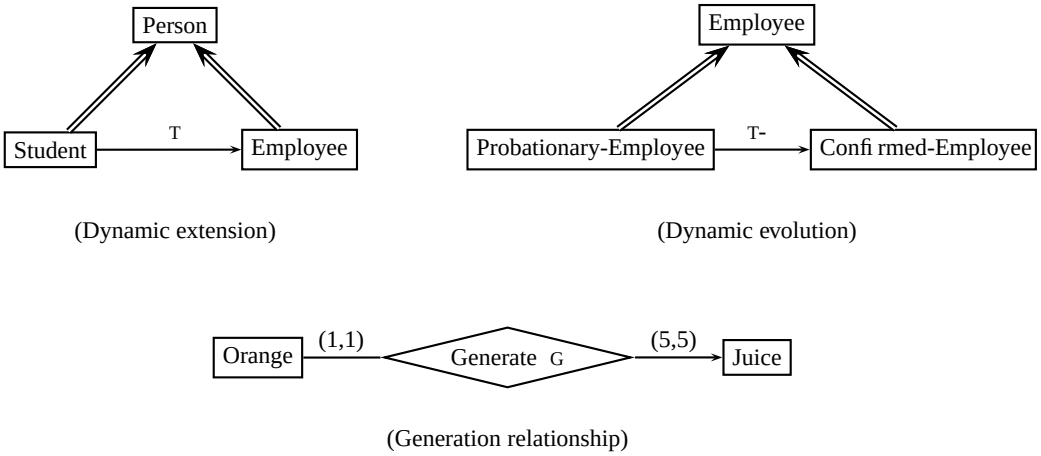

Figure 3: Dynamic relationships in MADS.

assuming that both `City` and `Street` follow the temporal behaviour of `Address` three different cases are possible as illustrated in Figure 2:

1. `Address` changes during time but the `Home-Phone` for a given address is always the same;
2. `Address` remains the same also if the associated `Home-Phone` can change over time (this is in agreement with the above mentioned identity principle);
3. Both `Address` and `Home-Phone` evolve over time—this being the most general situation.

In addition to both TIMEER and ERT, MADS models *dynamic* relationships. In a *transition* relationship the instances of an entity may eventually become instances of another entity. The instances of the source entity are said to *migrate* into the target entity and the phenomenon is called *object migration*. There are two types of transitions: *dynamic evolution* when objects cease to be instances of the source entity, and *dynamic extension* otherwise. In general, type constraints enforce that both the source and the target entity belong to the same generalisation hierarchy. Figure 3 shows an example of dynamic extension between the entities `Student` and `Employee`—represented with an arrows from the source to the target entity with the label T, and a case of dynamic evolution between a `Probationary-Employee` and a `Confirmed-Employee`—represented with an arrow from the source to the target entity with the label T-.

Another case of a dynamic relationship is the one called *generation* relationship. This relationship involves different instances—differently from the transition case: an instance (set of instances) from a source entity is (are) transformed in an instance of the target entity, and all the instances of the source are consumed in the transformation process. Figure 3 shows a relationship `Generate` between `Orange` and `Juice` marked with the label G and an arrow pointing to the target entity—the cardinality says that 5 oranges are needed to produce a single juice.

2.1 The Proposed model: $\mathcal{ER}_{VT}$

$\mathcal{ER}_{VT}$ is a temporal ER model that supports valid time for entities, relationships and attributes in the line of TIMEER and ERT, while supporting some form of dynamic relationships as introduced in MADS. The main motivations behind the development of $\mathcal{ER}_{VT}$ is the need for a formally specified temporal ER language with a clear semantics for the various temporal constructs, and the possibility to perform automatic deductions both on $\mathcal{ER}_{VT}$ schemas and on temporal queries posed against an $\mathcal{ER}_{VT}$ schema.

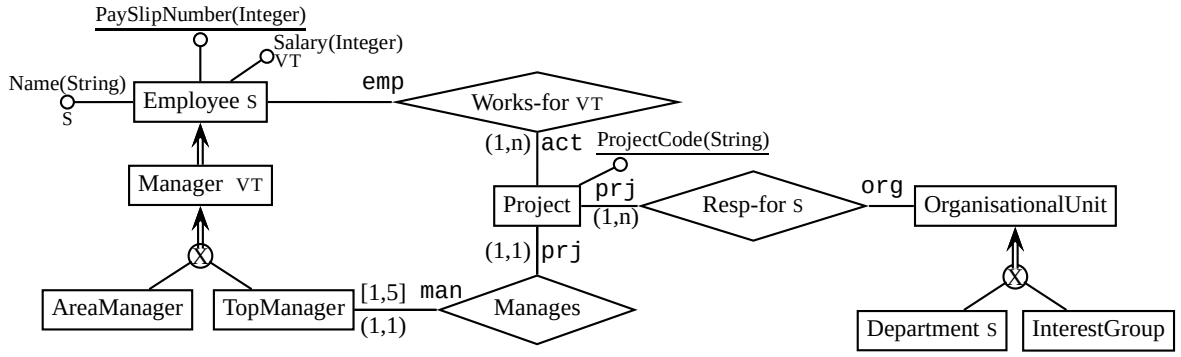

Figure 4: An $\mathcal{ER}_{VT}$ diagram.

The core model

We achieve temporal support in $\mathcal{ER}_{VT}$ by giving a temporal semantics to each ER construct—as in the implicit approach—and then adding new temporal constructs—as in the explicit approach. In this way the implicit and explicit approach are reconciled in a uniform framework where the second is built on top of the first. As a result we obtain an homogeneous language with a clear and elegant semantics.

The ability to add temporal constructs on top of a temporally implicit model has the advantage of preserving the non-temporal semantics of conventional (legacy) ER schemas when embedded into temporal ER diagrams, then $\mathcal{ER}_{VT}$ is *upward compatible*. The possibility of capturing the meanings of both legacy and temporal ER diagrams is crucial in modelling data warehouses or federated databases, where sources may be collections of both temporal and legacy databases. The formalisation we propose is in agreement with the desirable *orthogonality* principle [Spaccapietra *et al.*, 1998], since temporal constructs can be specified separately and independently for entities, relationships and attributes. Depending on the application requirements, the temporal support is decided by the designer. Furthermore, each $\mathcal{ER}_{VT}$ schema is *snapshot reducible* [Snodgrass, 1987], i.e., snapshot of the database described by an $\mathcal{ER}_{VT}$ schema is the same as the database described by the same schema where all the temporal constructs are eliminated and the schema is interpreted atemporally.

$\mathcal{ER}_{VT}$ can be seen as a temporal extension of the extended entity-relationship (EER) model[3]. Temporal explicit marks are added to capture the temporal behaviour. In particular, we suppose that entities, relationships and attributes in $\mathcal{ER}_{VT}$ can be either S-*marked* in what case they are considered *snapshot* constructs (i.e., each of their instances has a global lifetime, as in the case they derive from a legacy diagram), VT-*marked* and they are considered *temporary* constructs (i.e., each of their instances has a temporary existence), or *un-marked*, i.e. without any temporal mark, in what case they have temporally unconstrained instances (i.e., their instances can have either a global or a temporary existence). Cardinalities in $\mathcal{ER}_{VT}$ are distinguished between *snapshot participation constraints* (true at each point in time) and *lifespan participation constraints* (evaluated during the entire existence of the entity). Figure 4 shows how $\mathcal{ER}_{VT}$ models the temporal ER schema of the running example (Figure 1). It has to be noted that: *i.* there is no support for transaction time, *ii.* thanks to the combination between the implicit and explicit approach and the formal semantics associated to $\mathcal{ER}_{VT}$ (see Section 2.2) we don't need to specify all the temporal constraints since they are logically implied—e.g., since `Manager` is VT-marked then its sub-entities, left *un-marked*, `AreaManager` and `TopManager` are necessarily VT-marked, too.

[3]EER is the standard entity-relationship data model, enriched with ISA links, generalised hierarchies with disjoint and covering constraints, and full cardinality constraints [Elmasri and Navathe, 1994].

Advanced features

$\mathcal{ER}_{VT}$ can capture composite attributes. As far as the temporal properties of composite attributes is concerned, we adopt the same approach of MADS: components of a composite attribute can have a temporal behaviour different from the owner, respecting in this way the orthogonality principle. In particular, the diagrams of Figure 2 can be regarded as diagrams in $\mathcal{ER}_{VT}$, too—with the proviso of S-marking the unlabelled constructs.

$\mathcal{ER}_{VT}$ is able to capture some of the dynamic relationships introduced in MADS. It is possible to represent dynamic entities whose instances may change during time—both dynamic extension and evolution are captured. Generation relationships are also captured, then all the diagrams in Figure 3 are valid diagrams in $\mathcal{ER}_{VT}$, too. Furthermore, $\mathcal{ER}_{VT}$ is able to represent some form of schema evolution (for more details on the dynamic features in $\mathcal{ER}_{VT}$ see Section 5.3).

2.2 A formalisation of $\mathcal{ER}_{VT}$

In this paper, we adopt the *snapshot* representation of abstract temporal databases and temporal conceptual models (see e.g. [Chomicki and Toman, 1998]). 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 mapping from time points in $\mathcal{T}$ to standard relational databases, with the same interpretation of constants and the same domains along time. As it is well known, the snapshot model is in correspondence to the *timestamp* one, which is used, e.g., in the the bitemporal conceptual data model (*BCDM*) [Jensen *et al.*, 1994; Jensen and Snodgrass, 1999]. In the *BCDM* model each tuple in a bitemporal relation instance is associated with a bitemporal timestamp value modelling both the valid and the transaction time of the tuple.

The proposed formalisation is based on a linear syntax and an associated model-theoretic semantics as proposed in [Calvanese *et al.*, 1999] for the EER model, which is here extended to take into account the time dimension and the temporal constructs. This will give both a formal characterisation of the most important temporal conceptual modelling constructs (for the valid time representation) appeared in the literature [Gregersen and Jensen, 1998; 1999; Spaccapietra *et al.*, 1998; Theodoulidis *et al.*, 1991], and the formal background to develop the correspondence to the temporal description logic $\mathcal{DLR}_{\mathcal{US}}$ in order to reason over temporal schemas and queries. What follows is the formalisation of the core model of $\mathcal{ER}_{VT}$; composite attributes and the dynamic aspects will be formalised using directly the mapping into $\mathcal{DLR}_{\mathcal{US}}$. We adopt the following notation: given two sets X, Y a *X-labelled* tuple over Y is a function from X to Y; the labelled tuple T that maps the set $\{x_1, \ldots, x_n\} \subseteq X$ to the set $\{y_1, \ldots, y_n\} \subseteq y$ is denoted by $\langle x_1 : y_1, \ldots, x_n : y_n \rangle$, while $T[x_i] = y_i$.

Definition 2.1 ($\mathcal{ER}_{VT}$ Syntax). An $\mathcal{ER}_{VT}$ schema is a tuple $\Sigma_{\mathcal{S}} = (\mathcal{L}_{\mathcal{S}}, \text{REL}_{\mathcal{S}}, \text{ATT}_{\mathcal{S}}, \text{CARD}_{\mathcal{S}}, \text{CARD}^{L}_{\mathcal{S}}, \text{ISA}_{\mathcal{S}}, \text{DISJ}_{\mathcal{S}}, \text{COVER}_{\mathcal{S}}, \text{DISCOVER}_{\mathcal{S}}, \text{S}_{\mathcal{S}}, \text{T}_{\mathcal{S}}, \text{KEY}_{\mathcal{S}})$, where:

- $\mathcal{L}_{\mathcal{S}}$ is a finite alphabet partitioned into a set $\mathcal{E}$ of *entity* symbols, a set $\mathcal{A}_{\mathcal{S}}$ of *attribute* symbols, a set $\mathcal{R}_{\mathcal{S}}$ of *relationship* symbols, a set $\mathcal{U}_{\mathcal{S}}$ of *role* symbols, and a set $\mathcal{D}_{\mathcal{S}}$ of *domain* symbols. Furthermore, $\mathcal{E}_{\mathcal{S}}$ is partitioned into a set $\mathcal{E}^{S}_{\mathcal{S}}$ of *snapshot entities* (the S-*marked* entities in our graphical representation), a set $\mathcal{E}^{I}_{\mathcal{S}}$ of *temporal entities* (the *un-marked*, *I*mplicitly temporal, entities), and a set $\mathcal{E}^{T}_{\mathcal{S}}$ of *temporary entities* (the VT-*marked* entities). A similar partition applies to the set $\mathcal{R}_{\mathcal{S}}$.
- $\text{ATT}_{\mathcal{S}}$ is a function that maps each entity symbol in $\mathcal{E}_{\mathcal{S}}$ to an $\mathcal{A}_{\mathcal{S}}$-labelled tuple over $\mathcal{D}_{\mathcal{S}}$: $\text{ATT}_{\mathcal{S}}(E) = \langle A_1 : D_1, \ldots, A_h : D_h \rangle$.
- $\text{REL}_{\mathcal{S}}$ is a function that maps each relationship symbol in $\mathcal{R}_{\mathcal{S}}$ to an $\mathcal{U}_{\mathcal{S}}$-labelled tuple over $\mathcal{E}_{\mathcal{S}}$: $\text{REL}_{\mathcal{S}}(R) = \langle U_1 : E_1, \ldots, U_k : E_k \rangle$, and k is the *arity* of R.

- $\text{CARD}_{\mathcal{S}}$, and $\text{CARD}^{L}_{\mathcal{S}}$ are functions from $\mathcal{E}_{\mathcal{S}} \times \mathcal{R}_{\mathcal{S}} \times \mathcal{U}_{\mathcal{S}}$ to $\mathbb{N} \times (\mathbb{N} \cup \{\mathtt{N}\})$ such that if $\text{REL}_{\mathcal{S}}(R) = \langle U_1 : E_1, \ldots, U_k : E_k \rangle$ then $\text{CARD}_{\mathcal{S}}(E, R, U)$, $\text{CARD}^{L}_{\mathcal{S}}(E, R, U)$ are defined only if $U = U_i$ and $E = E_i$ for some $i \in \{1, \ldots, k\}$. We denote with $\text{CMIN}_{\mathcal{S}}(E, R, U), \text{CMAX}_{\mathcal{S}}(E, R, U)$ ($\text{CMIN}^{L}_{\mathcal{S}}(E, R, U), \text{CMAX}^{L}_{\mathcal{S}}(E, R, U)$) the first and second component of $\text{CARD}_{\mathcal{S}}$ ($\text{CARD}^{L}_{\mathcal{S}}$). If not stated otherwise, $\text{CMIN}_{\mathcal{S}}$ and $\text{CMIN}^{L}_{\mathcal{S}}$ are assumed to be 0 while $\text{CMAX}_{\mathcal{S}}$ and $\text{CMAX}^{L}_{\mathcal{S}}$ are assumed to be $\mathtt{N}$.

- $\text{ISA}_{\mathcal{S}} \subseteq (\mathcal{E}_{\mathcal{S}} \times \mathcal{E}_{\mathcal{S}}) \cup (\mathcal{R}_{\mathcal{S}} \times \mathcal{R}_{\mathcal{S}})$ is a binary relationship—$\text{ISA}_{\mathcal{S}}$ between relationships is restricted to relationships with the same arity.

- $\text{DISJ}_{\mathcal{S}}, \text{COVER}_{\mathcal{S}}, \text{DISCOVER}_{\mathcal{S}}$ are binary relations over $2^{\mathcal{E}}_{\mathcal{S}} \times \mathcal{E}_{\mathcal{S}}$.

- $\text{S}_{\mathcal{S}}, \text{T}_{\mathcal{S}}$ are binary relations over $\mathcal{E}_{\mathcal{S}} \times \mathcal{A}_{\mathcal{S}}$ containing, respectively, the snapshot and temporary attributes of an entity. Furthermore, if $\langle E, A \rangle \in \text{S}_{\mathcal{S}}, \text{T}_{\mathcal{S}}$ then A is between the attributes in $\text{ATT}_{\mathcal{S}}(E)$.

- $\text{KEY}_{\mathcal{S}}$ is a function that maps entity symbols in $\mathcal{E}_{\mathcal{S}}$ to attributes in $\mathcal{A}_{\mathcal{S}}$: $\text{KEY}_{\mathcal{S}}(E) = A$. Furthermore, if $\text{KEY}_{\mathcal{S}}(E) = A$ then A is between the attributes in $\text{ATT}_{\mathcal{S}}(E)$.

Example 2.2. The informal meaning of the various components of a schema is now illustrated with respect to the schema of Figure 4. The set of snapshot entities (relationships) is $\mathcal{E}^{S}_{\mathcal{S}} = \{\mathtt{Employee}, \mathtt{Department}\}$ ($\mathcal{R}^{S}_{\mathcal{S}} = \{\mathtt{Resp\text{-}for}\}$), the set of temporary entities (relationships) is $\mathcal{E}^{T}_{\mathcal{S}} = \{\mathtt{Manager}\}$ ($\mathcal{R}^{T}_{\mathcal{S}} = \{\mathtt{Works\text{-}for}\}$), the set of un-marked entities (relationships) is $\mathcal{E}^{I}_{\mathcal{S}} = \{\mathtt{AreaManager}, \mathtt{TopManager}, \mathtt{OrganisationalUnit}, \mathtt{InterestGroup}\}$ ($\mathcal{R}^{I}_{\mathcal{S}} = \{\mathtt{Manages}\}$). The function $\text{ATT}_{\mathcal{S}}$ is used to associate attributes to an entity, e.g., $\text{ATT}_{\mathcal{S}}(\mathtt{Employee}) = \langle \mathtt{PaySlipNumber} : \mathtt{Integer}, \mathtt{Salary} : \mathtt{Integer}, \mathtt{Name} : \mathtt{String} \rangle$. The function $\text{REL}_{\mathcal{S}}$ associates a name—called *role*—or, alternatively, a position number to each argument of a relationship, and for each role/position an entity, e.g., $\text{REL}_{\mathcal{S}}(\mathtt{Manages}) = \langle \mathtt{man} : \mathtt{TopManager}, \mathtt{prj} : \mathtt{Project} \rangle$. $\text{CARD}_{\mathcal{S}}, \text{CARD}^{L}_{\mathcal{S}}$ are used to associate snapshot and lifespan cardinality constraints, respectively, e.g., $\text{CARD}_{\mathcal{S}}(\mathtt{TopManager}, \mathtt{Manages}, \mathtt{man}) = \langle 1, 1 \rangle$, and $\text{CARD}^{L}_{\mathcal{S}}(\mathtt{TopManager}, \mathtt{Manages}, \mathtt{man}) = \langle 1, 5 \rangle$. $\text{ISA}_{\mathcal{S}}, \text{DISJ}_{\mathcal{S}}, \text{COVER}_{\mathcal{S}}, \text{DISCOVER}_{\mathcal{S}}$ are used for representing generalised hierarchies. $\text{ISA}_{\mathcal{S}}$ models the sub-entity relationship, e.g., $\mathtt{Manager}\text{ISA}_{\mathcal{S}}\mathtt{Employee}$ says that manager is a sub-entity of employee. $\text{DISJ}_{\mathcal{S}}$ models disjoint hierarchies, e.g., $\{\mathtt{AreaManager}, \mathtt{TopManager}\}\text{DISJ}_{\mathcal{S}}\mathtt{Manager}$ says that area manager is disjoint from top manager and both are sub-entities of manager. $\text{COVER}_{\mathcal{S}}$ models the fact that a set of sub-entities may have common instances but each instance of the super-entity belongs to at least one of those sub-entities. $\text{DISCOVER}_{\mathcal{S}}$ models the combination of disjoint and covering hierarchies, e.g. $\{\mathtt{AreaManager}, \mathtt{TopManager}\}\text{DISCOVER}_{\mathcal{S}}\mathtt{Manager}$. The relations $\text{S}_{\mathcal{S}}, \text{T}_{\mathcal{S}}$ model snapshot and temporary attributes, respectively, e.g., $\langle \mathtt{Employee}, \mathtt{Name} \rangle \in \text{S}_{\mathcal{S}}$, and $\langle \mathtt{Employee}, \mathtt{Salary} \rangle \in \text{T}_{\mathcal{S}}$. The function $\text{KEY}_{\mathcal{S}}$ captures keys for entities, e.g., the fact that the pay slip number is a key for an employee is captured by $\text{KEY}_{\mathcal{S}}(\mathtt{Employee}) = \mathtt{PaySlipNumber}$.

The model-theoretic semantics for the core model of $\mathcal{ER}_{VT}$ is given as an extension to the non temporal semantics introduced in [Calvanese *et al.*, 1998c].

Definition 2.3 ($\mathcal{ER}_{VT}$ Semantics). Let $\Sigma_{\mathcal{S}} = (\mathcal{L}_{\mathcal{S}}, \text{REL}_{\mathcal{S}}, \text{ATT}_{\mathcal{S}}, \text{CARD}_{\mathcal{S}}, \text{CARD}^{L}_{\mathcal{S}}, \text{ISA}_{\mathcal{S}}, \text{DISJ}_{\mathcal{S}}, \text{COVER}_{\mathcal{S}}, \text{DISCOVER}_{\mathcal{S}}, \text{S}_{\mathcal{S}}, \text{T}_{\mathcal{S}}, \text{KEY}_{\mathcal{S}})$ be an $\mathcal{ER}_{VT}$ schema, $BD = \bigcup_{D_i \in \mathcal{D}_{\mathcal{S}}} BD_i$ be a set of *basic domains* such that $BD_i \cap BD_j = \emptyset$ for $i \neq j$. $\mathcal{B} = (\mathcal{T}, \Delta^{\mathcal{B}} \cup \Delta^{\mathcal{B}}_{D}, \cdot^{\mathcal{B}(t)})$ is a *Temporal Database State* for the schema $\Sigma_{\mathcal{S}}$ where:

- $\Delta^{\mathcal{B}}$ is a non-empty set disjoint from $\Delta^{\mathcal{B}}_{D}$.

- $\Delta^{\mathcal{B}}_{D} = \bigcup_{D_i \in \mathcal{D}_{\mathcal{S}}} \Delta^{\mathcal{B}}_{D_i}$ is the set of basic domain values used in the schema $\Sigma_{\mathcal{S}}$ such that $\Delta^{\mathcal{B}}_{D_i} \subseteq BD_i$—we call $\Delta^{\mathcal{B}}_{D_i}$ *active domain*.

- $\cdot^{\mathcal{B}(t)}$ is a function that for each $t \in \mathcal{T}$ maps:

- Every domain symbol $D_i \in \mathcal{D}_\mathcal{S}$ to the corresponding active domain $D_i^{\mathcal{B}(t)} = \Delta_{D_i}^\mathcal{B}$—then $D_i^{\mathcal{B}(t)}$ doesn't depend from the time t of evaluation.
- Every entity $E \in \mathcal{E}_\mathcal{S}$ to a set $E^{\mathcal{B}(t)} \subseteq \Delta^\mathcal{B}$.
- Every relationship $R \in \mathcal{R}_\mathcal{S}$ to a set $R^{\mathcal{B}(t)}$ of $\mathcal{U}_\mathcal{S}$-labelled tuples over $\Delta^\mathcal{B}$.
- Every attribute $A \in \mathcal{A}_\mathcal{S}$ to a set $A^{\mathcal{B}(t)} \subseteq \Delta^\mathcal{B} \times \Delta_D^\mathcal{B}$.

$\mathcal{B}$ is a *legal temporal database state* if it satisfi es all the integrity constraints expressed in the schema:

(c1) For each $E_1, E_2 \in \mathcal{E}_\mathcal{S}$ if $E_1 \text{ISA}_\mathcal{S} E_2$ then, $\forall t \in \mathcal{T}.E_1^{\mathcal{B}(t)} \subseteq E_2^{\mathcal{B}(t)}$.

(c2) For each $R_1, R_2 \in \mathcal{R}_\mathcal{S}$ if $R_1 \text{ISA}_\mathcal{S} R_2$ then, $\forall t \in \mathcal{T}.R_1^{\mathcal{B}(t)} \subseteq R_2^{\mathcal{B}(t)}$.

(c3) For each $E \in \mathcal{E}_\mathcal{S}$ if $\text{ATT}_\mathcal{S}(E) = \langle A_1 : D_1, \ldots, A_h : D_h \rangle$ then, $\forall e \in E^{\mathcal{B}(t)}, \forall i \in \{1, \ldots, h\}, \exists! a_i.$ $\langle e, a_i \rangle \in A_i^{\mathcal{B}(t)} \wedge \forall a_i.\langle e, a_i \rangle \in A_i^{\mathcal{B}(t)} \rightarrow a_i \in \Delta_{D_i}^\mathcal{B}$—we consider, for simplicity, only single-valued attributes[4].

(c4) For each $R \in \mathcal{R}_\mathcal{S}$ if $rel(R) = \langle U_1 : E_1, \ldots, U_k : E_k \rangle$ then, $\forall r \in R^{\mathcal{B}(t)}.r = \langle U_1 : e_1, \ldots, U_k : e_k \rangle \wedge \forall i \in \{1, \ldots, k\}.e_i \in E_i^{\mathcal{B}(t)}$. In the following we adopt the convention: $\langle U_1 : e_1, \ldots, U_k : e_k \rangle \equiv \langle e_1, \ldots, e_k \rangle$, and $r[U_i] \equiv r[i]$ to denote the U_i/i-component of r—i.e., the *naming* and the *positional* notation for tuples are two equivalent notations.

(c5) For each cardinality constraint $\text{CARD}_\mathcal{S}(E, R, U)$ then, $\forall e \in E^{\mathcal{B}(t)}.\text{CMIN}_\mathcal{S}(E, R, U) \leq \#\{r \in R^{\mathcal{B}(t)} \mid r[U] = e\} \leq \text{CMAX}_\mathcal{S}(E, R, U)$.

(c6) For each lifespan cardinality constraint $\text{CARD}_\mathcal{S}^L(E, R, U)$ then, $\forall e \in E^{\mathcal{B}(t)}.\text{CMIN}_\mathcal{S}^L(E, R, U) \leq \#\bigcup_{t' \in \mathcal{T}}\{r \in R^{\mathcal{B}(t')} \mid r[U] = e\} \leq \text{CMAX}_\mathcal{S}^L(E, R, U)$.

(c7) For each snapshot entity $E \in \mathcal{E}_\mathcal{S}^S$ then, $e \in E^{\mathcal{B}(t)} \rightarrow (\forall t' \in \mathcal{T}.e \in E^{\mathcal{B}(t')})$.

(c8) For each temporary entity $E \in \mathcal{E}_\mathcal{S}^T$ then, $e \in E^{\mathcal{B}(t)} \rightarrow (\exists t_1 > t.e \notin E^{\mathcal{B}(t_1)} \vee \exists t_2 < t.e \notin E^{\mathcal{B}(t_2)})$.

(c9) For each snapshot relationship $R \in \mathcal{R}_\mathcal{S}^S$ then, $r \in R^{\mathcal{B}(t)} \rightarrow (\forall t' \in \mathcal{T}.r \in R^{\mathcal{B}(t')})$.

(c10) For each temporary relationship $R \in \mathcal{R}_\mathcal{S}^T$ then, $r \in R^{\mathcal{B}(t)} \rightarrow (\exists t_1 > t.r \notin R^{\mathcal{B}(t_1)} \vee \exists t_2 < t.r \notin R^{\mathcal{B}(t_2)})$.

(c11) For each entity $E \in \mathcal{E}_\mathcal{S}$ if $\text{ATT}_\mathcal{S}(E) = \langle A_1 : D_1, \ldots, A_h : D_h \rangle$, and $\langle E, A_i \rangle \in \text{S}_\mathcal{S}$ then, $\forall e \in E^{\mathcal{B}(t)}.\langle e, a_i \rangle \in A_i^{\mathcal{B}(t)} \rightarrow \forall t' \in \mathcal{T}.\langle e, a_i \rangle \in A_i^{\mathcal{B}(t')}$.

(c12) For each entity $E \in \mathcal{E}_\mathcal{S}$ if $\text{ATT}_\mathcal{S}(E) = \langle A_1 : D_1, \ldots, A_h : D_h \rangle$, and $\langle E, A_i \rangle \in \text{T}_\mathcal{S}$ then, $\forall e \in E^{\mathcal{B}(t)}.\langle e, a_i \rangle \in A_i^{\mathcal{B}(t)} \rightarrow \exists t' \neq t.\langle e, a_i \rangle \notin A_i^{\mathcal{B}(t')}$.

(c13) For $E, E_1, \ldots, E_n \in \mathcal{E}_\mathcal{S}$ if $\{E_1, \ldots, E_n\} \text{DISJ}_\mathcal{S} E$ then, $\forall i \in \{1, \ldots, n\}.E_i \text{ISA}_\mathcal{S} E \wedge \forall t \in \mathcal{T}, \forall j \in \{1, \ldots, n\}, j \neq i.E_i^{\mathcal{B}(t)} \cap E_j^{\mathcal{B}(t)} = \emptyset$.

(c14) For $E, E_1, \ldots, E_n \in \mathcal{E}_\mathcal{S}$ if $\{E_1, \ldots, E_n\} \text{COVER}_\mathcal{S} E$ then, $\forall i \in \{1, \ldots, n\}.E_i \text{ISA}_\mathcal{S} E \wedge \forall t \in \mathcal{T}.E^{\mathcal{B}(t)} = \bigcup_{i=1}^n E_i^{\mathcal{B}(t)}$.

(c15) For $E, E_1, \ldots, E_n \in \mathcal{E}_\mathcal{S}$ if $\{E_1, \ldots, E_n\} \text{DISCOVER}_\mathcal{S} E$ then, $\{E_1, \ldots, E_n\} \text{DISJ}_\mathcal{S} E$ and $\{E_1, \ldots, E_n\}$ $\text{COVER}_\mathcal{S} E$.

[4] Section 5.2 shows that $\mathcal{ER}_{VT}$ supports also multi-valued attributes.

(c16) For each $E \in \mathcal{E}_\mathcal{S}, A \in \mathcal{A}_\mathcal{S}$ such that $\text{KEY}_\mathcal{S}(E) = A$ then the same semantic equation of **c11** is true—i.e., a key is a snapshot attribute—and $\forall a \in \Delta_D^\mathcal{B}.\#\{e \in E^{\mathcal{B}(t)} \mid \langle e, a\rangle \in A^{\mathcal{B}(t)}\} \leq 1$.

Reasoning over a temporal schema includes verifying whether an entity, relationship or a schema admit a non empty interpretation, and checking for sub-entities and sub-relationships.

Definition 2.4 (Reasoning in $\mathcal{ER}_{VT}$). Let $\Sigma_\mathcal{S}$ an $\mathcal{ER}_{VT}$ schema, $E, E' \in \mathcal{E}_\mathcal{S}$ and $R, R' \in \mathcal{R}_\mathcal{S}$. The following are the reasoning services over $\Sigma_\mathcal{S}$:

1. E (R) is *satisfiable* if there exists a legal temporal database state $\mathcal{B}$ for $\Sigma_\mathcal{S}$ such that $E^{\mathcal{B}(t)} \neq \emptyset$ ($R^{\mathcal{B}(t)} \neq \emptyset$), for some $t \in \mathcal{T}$;
2. E (R) is *globally satisfiable* if there exists a legal temporal database state $\mathcal{B}$ for $\Sigma_\mathcal{S}$ such that $E^{\mathcal{B}(t)} \neq \emptyset$ ($R^{\mathcal{B}(t)} \neq \emptyset$), for all $t \in \mathcal{T}$;
3. E (R) is a *sub-entity* of E' (R') if for all possible legal temporal database states $\mathcal{B}$ for $\Sigma_\mathcal{S}$ and for all $t \in \mathcal{T}$: $E^{\mathcal{B}(t)} \subseteq E'^{\mathcal{B}(t)}$ ($R^{\mathcal{B}(t)} \subseteq R'^{\mathcal{B}(t)}$);
4. $\Sigma_\mathcal{S}$ is *satisfiable* if there exists a legal temporal database state $\mathcal{B}$ for $\Sigma_\mathcal{S}$ such that all the entities and relationships are satisfiable at some fixed $t \in \mathcal{T}$;
5. $\Sigma_\mathcal{S}$ is *globally satisfiable* if there exists a legal temporal database state $\mathcal{B}$ for $\Sigma_\mathcal{S}$ such that all the entities and relationships are globally satisfiable.
6. ...

By considering (temporal) ER schemas as a set of constraints we are mainly interested in checking whether such constraints admit a model, i.e., whether there could be a database state satisfying them. In the temporal database community the notion of *potential constraint satisfaction*[Chomicki and Toman, 1998] has been also defined. Given a current (finite) history of a database, a constraint can potentially be satisfied if there is a model for it that extends the current history. This paper does not deal with potential constraint satisfaction that will be matter of future work.

The following 'classical' desirable features in temporal conceptual modelling come as almost trivial consequences in our framework, and if not explicitly stated are logically implied.

Proposition 2.5. *In every $\mathcal{ER}_{VT}$ schema the following temporal properties hold:*

1. *Sub-entities of temporary entities are also temporary.*
2. *Sub-entities of snapshot entities, and super-entities of temporary or un-marked entities can be either snapshot, temporary or un-marked entities.*
3. *Super-entities of snapshot entities are also snapshot.*
4. *A schema is inconsistent if exactly one of a whole set of snapshot partitioning sub-entities is temporary.*
5. *Participants of snapshot relations are either snapshot or un-marked entities. They are snapshot when they participate at least once in the relationship.*
6. *Participants of temporary or un-marked relations can be either snapshot, temporary or un-marked entities.*
7. *A relationship is temporary if one of the participating entities is temporary.*

8. *The temporal behaviour for an entity is independent from that of its attributes.*

Points $1-3$ *are true also for relationships.*

Example 2.6. From the $\mathcal{ER}_{VT}$ diagram in Fig. 4 the following logical implications hold:

1. Since `Manager` is a temporary entity then both `AreaManager` and `TopManager` are temporary entities—constraining either `AreaManager` or `TopManager` as snapshot entities would lead to a contradiction.
2. Even if `Employee` is a snapshot entity it is consistent to have `Manager`—a temporary entity—as a sub-entity of `Employee`.
3. The fact that `OrganisationalUnit` is necessarily snapshot is a valid logical implication since it is a super-entity of `Department`—a snapshot entity.
4. The fact that `InterestGroup` is a snapshot entity follows logically from our theory.
5. Since `Project` participates at least once in the snapshot relationship `Resp-For` it must be a snapshot entity.
6. The un-marked relationship `Manages` is consistent even if the snapshot entity `Project` participates in the relationship.
7. The fact that `Manages` is a temporary relationship follows logically from our theory since the temporary entity `TopManager` participates in the relationship.

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

As a logic language for reasoning over temporal database conceptual schemas we use a natural combination of the propositional linear temporal logic with *Since* and *Until*[Sistla and Clarke, 1985; Gabbay *et al.*, 1994] and the (non-temporal) description logic $\mathcal{DLR}$ [Calvanese *et al.*, 1998a]. 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 ≥ 2. Starting from a set of *atomic entities* (denoted by EN), a set of *atomic relations* (denoted by RN) and a set of *role symbols* (denoted by U) we define inductively (complex) entity and relation expressions as it is shown in the upper part of Fig. 5, 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.

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 $U_i/n : E$ denotes an n-ary relation whose argument named U_i ($i \leq n$) is of type E; if it is clear from the context, we omit n and write $U_i : E$. The projection expression $\exists^{\leqslant k}[U_i]R$ is a generalisation with cardinalities of the projection operator over the argument named U_i of the relation R—this last operator coincides with $\exists^{\geqslant 1}[U_j]R$. It is also possible to use the argument position numbers version of the model by replacing role symbols with position numbers—i.e., $U_i/n : E \equiv i/n : E$, and $\exists^{\leqslant k}[U_j]R \equiv \exists^{\leqslant k}[j]R$.

A *knowledge base* is a finite set Σ of $\mathcal{DLR}_{\mathcal{US}}$-formulas. Atomic formulas are formulas of the form $E_1 \sqsubseteq E_2$ and $R_1 \sqsubseteq R_2$, with R_1 and R_2 being relations of the same arity. If φ and ψ are $\mathcal{DLR}_{\mathcal{US}}$-formulas, then so are $\neg\varphi, \varphi \wedge \psi, \varphi\, \mathcal{U}\, \psi, \varphi\, \mathcal{S}\, \psi$. $E_1 \doteq E_2$ is used as an abbreviation for $(E_1 \sqsubseteq E_2) \wedge (E_2 \sqsubseteq E_1)$, while $E_1 \sqsubseteq^* E_2$ is a *global atomic formula* and is a shortcut for $\Box^*(E_1 \sqsubseteq E_2)$—the same holds true for relations. Knowledge bases will serve to capture the constraints specified by a temporal database schema as will be clear in Section 4.

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

$$
\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)} \\
(U_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, t). \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}[U_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 5: Syntax and semantics of $\mathcal{DLR}_{\mathcal{US}}$.

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. 5, 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.[5] 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.

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

[5] 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 $\langle d_1, d_2 \rangle \notin (\top_2)^{\mathcal{I}(t+1)}$.

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

A formula φ 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 Σ is *satisfiable* if the conjunction $\bigwedge \Sigma$ of all formulas in Σ 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 Σ. We say that Σ 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 Σ (*globally*) *implies* φ 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 Σ is globally satisfiable iff $\boxplus(\bigwedge \Sigma)$ 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. [Chomicki and Toman, 1998; Hodkinson *et al.*, 2000] and Section 7 below. The expressive capabilities of $\mathcal{DLR}_{\mathcal{US}}$ for temporal conceptual modelling in databases are illustrated in Section 4.

4 Encoding $\mathcal{ER}_{VT}$ in $\mathcal{DLR}_{\mathcal{US}}$

In this section we show how the temporal description logic $\mathcal{DLR}_{\mathcal{US}}$ is able to capture temporal conceptual schema expressed in $\mathcal{ER}_{VT}$. This characterisation allows us to define and support several forms of reasoning on both temporal conceptual models and temporal queries by using the reasoning services of $\mathcal{DLR}_{\mathcal{US}}$. The correspondence is based on a mapping function Φ—introduced by [Calvanese *et al.*, 1998c; 1999] for capturing non-temporal ER models—from $\mathcal{ER}_{VT}$ schemas to $\mathcal{DLR}_{\mathcal{US}}$ knowledge bases.

Informally, the encoding works as follows. Entity and relationship symbols in the $\mathcal{ER}_{VT}$ diagram are mapped into $\mathcal{DLR}_{\mathcal{US}}$ entity and relation names. Domain symbols are mapped into additional entity names, pairwise disjoint. Attributes of entities are mapped to binary relations with number restrictions stating the single-valuedness[6]. ISA links between entities or between relationships are mapped using atomic formulas. Generalised hierarchies with disjointness and covering constraints can be captured using the Boolean connectives. Cardinality constraints are mapped using the number restriction quantifiers in $\mathcal{DLR}_{\mathcal{US}}$. Temporal properties in $\mathcal{ER}_{VT}$ are mapped using the temporal operators in $\mathcal{DLR}_{\mathcal{US}}$.

Definition 4.1 (Mapping $\mathcal{ER}_{VT}$ into $\mathcal{DLR}_{\mathcal{US}}$). Let $\Sigma_{\mathcal{S}} = (\mathcal{L}_{\mathcal{S}}, \text{REL}_{\mathcal{S}}, \text{ATT}_{\mathcal{S}}, \text{CARD}_{\mathcal{S}}, \text{CARD}_{\mathcal{S}}^{L}, \text{ISA}_{\mathcal{S}}, \text{DISJ}_{\mathcal{S}}, \text{COVER}_{\mathcal{S}}, \text{DISCOVER}_{\mathcal{S}}, \text{S}_{\mathcal{S}}, \text{T}_{\mathcal{S}}, \text{KEY}_{\mathcal{S}})$ be an $\mathcal{ER}_{VT}$ schema. The $\mathcal{DLR}_{\mathcal{US}}$ knowledge base $\Phi(\Sigma_{\mathcal{S}}) = (\mathit{EN}, \mathit{RN}, \mathit{RS}, \Sigma)$ is defined as follows.
The set EN of atomic entities is such that:

- For each domain symbol $D \in \mathcal{D}_{\mathcal{S}}$ then $\Phi(D) \in \mathit{EN}$;
- For each entity symbol $E \in \mathcal{E}_{\mathcal{S}}$ then $\Phi(E) \in \mathit{EN}$.

The set RN of atomic relationships is such that:

- For each relationship $R \in \mathcal{R}_{\mathcal{S}}$ then $\Phi(R) \in \mathit{RN}$;

[6] Multi-valued attributes can be easily captured by eliminating such cardinality constraint.

- For each attribute $A \in \mathcal{A}_\mathcal{S}$ then $\Phi(A) \in RN$.

The set RS of role symbols is such that:

- For each role symbol $U \in \mathcal{U}_\mathcal{S}$ then $\Phi(U) \in RS$;
- Two distinguished role symbols $\texttt{From}$ and $\texttt{To} \in RS$.

Φ is functional over $\mathcal{D}_\mathcal{S} \cup \mathcal{E}_\mathcal{S} \cup \mathcal{R}_\mathcal{S} \cup \mathcal{U}_\mathcal{S} \cup \mathcal{A}_\mathcal{S}$. The set Σ contains the following $\mathcal{DLR}_{\mathcal{US}}$ formulas—i.e., *temporal integrity constraints*.

(f1) For each $D_i \in \mathcal{D}_\mathcal{S}$:
$\Phi(D_i) \sqsubseteq^* (\Box^+\Phi(D_i)) \sqcap (\Box^-\Phi(D_i))$—i.e., $\Phi(D_i) \doteq \Box^*\Phi(D_i)$.

(f2) For each relationship $R \in \mathcal{R}_\mathcal{S}$ such that $\text{REL}_\mathcal{S}(R) = \langle U_1 : E_1, \ldots, U_k : E_k \rangle$:
$\Phi(R) \sqsubseteq^* \Phi(U_1)/k : \Phi(E_1) \sqcap \ldots \Phi(U_k)/k : \Phi(E_k)$.

(f3) For each attribute $A \in \mathcal{A}_\mathcal{S}$:
$\Phi(A) \sqsubseteq^* \texttt{From}/2 : \top \sqcap \texttt{To}/2 : \top$.

(f4) For each entity $E \in \mathcal{E}_\mathcal{S}$ such that $\text{ATT}_\mathcal{S}(E) = \langle A_1 : D_1, \ldots, A_h : D_h \rangle$:
$\Phi(E) \sqsubseteq^* \exists^{=1}[\texttt{From}]\Phi(A_1) \sqcap \exists^{=1}[\texttt{From}](\Phi(A_1) \sqcap \texttt{To}/2 : \Phi(D_1)) \sqcap \ldots \sqcap$
$\exists^{=1}[\texttt{From}]\Phi(A_h) \sqcap \exists^{=1}[\texttt{From}](\Phi(A_h) \sqcap \texttt{To}/2 : \Phi(D_h))$.

(f5) For each role symbol $U_i \in \mathcal{U}_\mathcal{S}$ between $R \in \mathcal{R}_\mathcal{S}$ and $E \in \mathcal{E}_\mathcal{S}$:

a. If $m = \text{CMIN}_\mathcal{S}(E, R, U_i) \neq 0$: $\Phi(E) \sqsubseteq^* \exists^{\geq m}[\Phi(U_i)]\Phi(R)$;

b. If $n = \text{CMAX}_\mathcal{S}(E, R, U_i) \neq \texttt{N}$: $\Phi(E) \sqsubseteq^* \exists^{\leq n}[\Phi(U_i)]\Phi(R)$.

(f6) For each role symbol $U_i \in \mathcal{U}_\mathcal{S}$ between $R \in \mathcal{R}_\mathcal{S}$ and $E \in \mathcal{E}_\mathcal{S}$:

a. If $m = \text{CMIN}^L_\mathcal{S}(E, R, U_i) \neq 0$: $\Phi(E) \sqsubseteq^* \exists^{\geq m}[\Phi(U_i)](\Diamond^*\Phi(R))$;

b. If $n = \text{CMAX}^L_\mathcal{S}(E, R, U_i) \neq \texttt{N}$: $\Phi(E) \sqsubseteq^* \exists^{\leq n}[\Phi(U_i)](\Diamond^*\Phi(R))$.

(f7) For each pair of entities (relationships) $E_1, E_2 \in \mathcal{E}_\mathcal{S}$ ($R_1, R_2 \in \mathcal{R}_\mathcal{S}$) such that $E_1 \text{ISA}_\mathcal{S} E_2$ ($R_1 \text{ISA}_\mathcal{S} R_2$):

a. $\Phi(E_1) \sqsubseteq^* \Phi(E_2)$;

b. $\Phi(R_1) \sqsubseteq^* \Phi(R_2)$.

(f8) For each snapshot entity $E \in \mathcal{E}^S_\mathcal{S}$:
$\Phi(E) \sqsubseteq^* (\Box^+\Phi(E)) \sqcap (\Box^-\Phi(E))$—i.e., $\Phi(E) \doteq \Box^*\Phi(E)$.

(f9) For each snapshot relationship $R \in \mathcal{R}^S_\mathcal{S}$:
$\Phi(R) \sqsubseteq^* (\Box^+\Phi(R)) \sqcap (\Box^-\Phi(R))$—i.e., $\Phi(R) \doteq \Box^*\Phi(R)$.

(f10) For each snapshot attribute A_i with $\langle E, A_i \rangle \in \text{S}_\mathcal{S}$:
$\Phi(E) \sqsubseteq^* \exists^{=1}[\texttt{From}](\Box^*\Phi(A_i))$.

(f11) For each temporary entity $E \in \mathcal{E}^T_\mathcal{S}$:
$\Phi(E) \sqsubseteq^* (\Diamond^+\neg\Phi(E)) \sqcup (\Diamond^-\neg\Phi(E))$.

(f12) For each temporary relationship $R \in \mathcal{R}^T_\mathcal{S}$:
$\Phi(R) \sqsubseteq^* (\Diamond^+\neg\Phi(R)) \sqcup (\Diamond^-\neg\Phi(R))$.

(f13) For each temporary attribute A_i with $\langle E, A_i \rangle \in \text{T}_\mathcal{S}$:
$\Phi(E)\rangle \sqsubseteq^* \exists^{=1}[\texttt{From}](\Phi(A_i) \sqcap (\Diamond^+\neg\Phi(A_i) \sqcup \Diamond^-\neg\Phi(A_i)))$.

(f14) For each pair of symbols $X_1, X_2 \in \mathcal{E}_\mathcal{S} \cup \mathcal{D}_\mathcal{S}$ such that $X_1 \neq X_2$ and $X_1 \in \mathcal{D}_\mathcal{S}$:
$\Phi(X_1) \sqsubseteq^* \neg\Phi(X_2)$.

(f15) For $E, E_1, \ldots, E_n \in \mathcal{E}_\mathcal{S}$ if $\{E_1, \ldots, E_n\}\text{DISJ}_\mathcal{S}E$:
$E_1 \sqsubseteq^* E \sqcap \neg E_1 \sqcap \ldots \sqcap \neg E_n$
$E_2 \sqsubseteq^* E \sqcap \neg E_3 \sqcap \ldots \sqcap \neg E_n$
$\ldots$
$E_n \sqsubseteq^* E$.

(f16) For $E, E_1, \ldots, E_n \in \mathcal{E}_\mathcal{S}$ if $\{E_1, \ldots, E_n\}\text{COVER}_\mathcal{S}E$:
$E_1 \sqsubseteq^* E$
$\ldots$
$E_n \sqsubseteq^* E$
$E \sqsubseteq^* E_1 \sqcup \ldots \sqcup E_n$.

(f17) For $E, E_1, \ldots, E_n \in \mathcal{E}_\mathcal{S}$ if $\{E_1, \ldots, E_n\}\text{DISCOVER}_\mathcal{S}E$ then we have all the formulas in `f15` and the formula: $E \sqsubseteq^* E_1 \sqcup \ldots \sqcup E_n$.

(f18) For each key attribute A with $\text{KEY}_\mathcal{S}(E) = A$:
$\Phi(E) \sqsubseteq^* \quad \exists^{=1}[\texttt{From}](\Box^*\Phi(A))$
$\top \sqsubseteq^* \quad \exists^{\leq 1}[\texttt{To}](\Phi(A) \sqcap \texttt{From}/2 : \Phi(E))$

Example 4.2. We now show how the schema of the running example of Figure 4 is translated. Let $\Sigma_\mathcal{S}^{ex}$ be the schema associated to the ER diagram of Figure 4 as illustrated in Example 2.2, and $\Phi(\Sigma_\mathcal{S}^{ex}) = (EN^{ex}, RN^{ex}, AN^{ex}, \Sigma^{ex})$ be its $\mathcal{DLR}_{\mathcal{US}}$ translation. The alphabet of $\Phi(\Sigma_\mathcal{S}^{ex})$ is such that:

$$\begin{array}{rcl} EN^{ex} & = & \{\texttt{Employee}, \texttt{Manager}, \texttt{AreaManager}, \texttt{TopManager}, \texttt{Project}, \texttt{OrganisationalUnit}, \\ & & \texttt{Department}, \texttt{InterestGroup}, \texttt{Integer}, \texttt{String}\} \\ RN^{ex} & = & \{\texttt{Manages}, \texttt{Works-for}, \texttt{Resp-for}\} \\ AN^{ex} & = & \{\texttt{PaySlipNumber}, \texttt{Salary}, \texttt{Name}, \texttt{ProjectCode}\} \end{array}$$

One of the main reasons for presenting the $\mathcal{ER}_{VT}$ model is that all the constructs in this model have a model-theoretic semantics and a mapping in the logic $\mathcal{DLR}_{\mathcal{US}}$ which allows us to use the reasoning services in $\mathcal{DLR}_{\mathcal{US}}$ to support several form of reasoning on conceptual schemas expressed in $\mathcal{ER}_{VT}$.

To prove that reasoning on $\mathcal{ER}_{VT}$ can be done by reasoning on its $\mathcal{DLR}_{\mathcal{US}}$ translation we need to prove the correctness of the encoding. The encoding has been proven correct for the ER model with implicit time in [Artale and Franconi, 1999a; 1999b], where $\mathcal{DLR}$ was interpreted by a temporal semantics. Here we give a proof for the full mapping considering both implicit and explicit temporal constructs. The mapping provided in Definition 4.1 is proved correct by establishing a precise correspondence between legal database states of $\mathcal{ER}_{VT}$ schemas and models of the corresponding $\mathcal{DLR}_{\mathcal{US}}$ theories.

Proposition 4.3 (Correctness of the encoding). *Let* $\Sigma_\mathcal{S} = (\mathcal{L}_\mathcal{S}, \text{REL}_\mathcal{S}, \text{ATT}_\mathcal{S}, \text{CARD}_\mathcal{S}, \text{CARD}_\mathcal{S}^L, \text{ISA}_\mathcal{S}, \text{DISJ}_\mathcal{S}, \text{COVER}_\mathcal{S}, \text{DISCOVER}_\mathcal{S}, \text{S}_\mathcal{S}, \text{T}_\mathcal{S}, \text{KEY}_\mathcal{S})$ *be an* $\mathcal{ER}_{VT}$ *schema, then:*

1. *For each legal temporal database state* $\mathcal{B}$ *for* $\Sigma_\mathcal{S}$ *there is a temporal model* $\mathcal{I}$ *of* $\Phi(\Sigma_\mathcal{S})$ *such that for each symbol* $X \in \mathcal{E}_\mathcal{S} \cup \mathcal{A}_\mathcal{S} \cup \mathcal{R}_\mathcal{S} \cup \mathcal{D}_\mathcal{S}$ *then* $\Phi(X)^{\mathcal{I}(t)} = X^{\mathcal{B}(t)}$*, for each* $t \in \mathcal{T}$.

2. *For each temporal model* $\mathcal{I} = \langle \mathcal{T}, \Delta^\mathcal{I}, \cdot^{\mathcal{I}(t)} \rangle$ *of* $\Phi(\Sigma_\mathcal{S})$ *there is a legal temporal database state* $\mathcal{B}$ *for* $\Sigma_\mathcal{S}$*, a set of basic domains* BD*, and a one-to-one partial function* $\mathcal{B}_\Delta : \Delta^\mathcal{I} \rightarrow BD$*—total on* $\bigcup_{D_i \in \mathcal{D}_\mathcal{S}} \Phi(D_i)^{\mathcal{I}(t)}$*—such that:*

   (a) *For each symbol* $X \in \mathcal{E}_\mathcal{S} \cup \mathcal{R}_\mathcal{S}$*:* $X^{\mathcal{B}(t)} = \Phi(X)^{\mathcal{I}(t)}$*, for each* $t \in \mathcal{T}$*;*

   (b) *For each* $D_i \in \mathcal{D}_\mathcal{S}$*:* $D_i^{\mathcal{B}(t)} = \mathcal{B}_\Delta(\Phi(D_i)^{\mathcal{I}(t)})$*, for each* $t \in \mathcal{T}$*;*

(c) For each $A \in \mathcal{A}_\mathcal{S}$: $\langle d_1, d_2 \rangle \in A^{\mathcal{I}(t)}$ iff $\langle d_1, \mathcal{B}_\Delta(d_2) \rangle \in A^{\mathcal{B}(t)}$, for each $t \in \mathcal{T}$.

As a direct consequence of the proposition 4.3 easily follows the theorem that reduces reasoning over $\mathcal{ER}_{VT}$ schemas (Definition 2.4) to reasoning over $\mathcal{DLR}_{\mathcal{US}}$ knowledge bases (note that $\models$ stands for *global* implication).

Theorem 4.4 (Reasoning over $\mathcal{ER}_{VT}$ schemas). *Let $\Sigma_\mathcal{S}$ be an $\mathcal{ER}_{VT}$ schema, $E, E' \in \mathcal{E}_\mathcal{S}$, $R, R' \in \mathcal{R}_\mathcal{S}$, and $\Phi(\Sigma_\mathcal{S})$ the $\mathcal{DLR}_{\mathcal{US}}$ knowledge base corresponding to $\Sigma_\mathcal{S}$. The following holds:*

1. *E (R) is satisfiable iff $\Phi(\Sigma_\mathcal{S}) \models \Diamond^* \neg(\Phi(E) \sqsubseteq \bot)$—$\Phi(\Sigma_\mathcal{S}) \models \Diamond^* \neg(\Phi(R) \sqsubseteq \bot)$;*
2. *E (R) is globally satisfiable iff $\Phi(\Sigma_\mathcal{S}) \models \neg(\Phi(E) \sqsubseteq \bot)$—$\Phi(\Sigma_\mathcal{S}) \models \neg(\Phi(R) \sqsubseteq \bot)$;*
3. *E (R) is a sub-entity of E' (R') iff $\Phi(\Sigma_\mathcal{S}) \models E \sqsubseteq E'$—$\Phi(\Sigma_\mathcal{S}) \models R \sqsubseteq R'$;*
4. *$\Sigma_\mathcal{S}$ is satisfiable iff $\Phi(\Sigma_\mathcal{S})$ is satisfiable;*
5. *$\Sigma_\mathcal{S}$ is globally satisfiable iff $\Phi(\Sigma_\mathcal{S})$ is globally satisfiable.*

5 Conceptual modelling in $\mathcal{DLR}_{\mathcal{US}}$

This Section illustrates the formulas contained in the $\mathcal{DLR}_{\mathcal{US}}$ knowledge base Σ^{ex} in order to capture the (temporal) integrity constraints specified by the $\mathcal{ER}_{VT}$ schema $\Sigma^{ex}_\mathcal{S}$.

5.1 The implicit entity-relationship model

$\mathcal{ER}_{VT}$ models temporal schemas by giving a temporal semantics to each ER construct and then using temporal integrity constraints to model explicit temporal constructs. As we said, $\mathcal{ER}_{VT}$ reconciles the implicit and the explicit approach. Here we illustrate those formulas in Σ^{ex} constructed using $\mathcal{DLR}$—the non-temporal fragment of $\mathcal{DLR}_{\mathcal{US}}$—that translate entities, relationships and attributes in the implicit style.

The relationships in $\Sigma^{ex}_\mathcal{S}$ give rise to the following formulas:

$$\begin{array}{rcl}
\texttt{Works-for} & \sqsubseteq^* & \texttt{emp}/2 : \texttt{Employee} \sqcap \texttt{act}/2 : \texttt{Project} \\
\texttt{Manages} & \sqsubseteq^* & \texttt{man}/2 : \texttt{TopManager} \sqcap \texttt{prj}/2 : \texttt{Project} \\
\texttt{Resp-for} & \sqsubseteq^* & \texttt{prj}/2 : \texttt{Project} \sqcap \texttt{org}/2 : \texttt{OrganisationalUnit}
\end{array}$$

The attributes for the entities in $\Sigma^{ex}_\mathcal{S}$ give rise to the following formulas:

$$\begin{array}{rcl}
\texttt{Employee} & \sqsubseteq^* & \exists^{=1}[\texttt{From}]\texttt{PaySlipNumber} \sqcap \exists^{=1}[\texttt{From}](\texttt{PaySlipNumber} \sqcap \texttt{To}/2 : \texttt{Integer}) \sqcap \\
 & & \exists^{=1}[\texttt{From}]\texttt{Salary} \sqcap \exists^{=1}[\texttt{From}](\texttt{Salary} \sqcap \texttt{To}/2 : \texttt{Integer}) \sqcap \\
 & & \exists^{=1}[\texttt{From}]\texttt{Name} \sqcap \exists^{=1}[\texttt{From}](\texttt{Name} \sqcap \texttt{To}/2 : \texttt{String}) \\
\texttt{Project} & \sqsubseteq^* & \exists^{=1}[\texttt{From}]\texttt{ProjectCode} \sqcap \exists^{=1}[\texttt{From}](\texttt{ProjectCode} \sqcap \texttt{To}/2 : \texttt{String})
\end{array}$$

where, for each attribute, the first conjunct declares the attribute as a functional—i.e., single valued—attribute while the second conjunct restricts the value of the attribute to belong to the appropriate basic domain.

Generalised hierarchies in $\Sigma^{ex}_\mathcal{S}$ give rise to the following formulas:

$$\begin{array}{rcl}
\texttt{Manager} & \sqsubseteq^* & \texttt{Employee} \sqcap (\texttt{AreaManager} \sqcup \texttt{TopManager}) \\
\texttt{AreaManager} & \sqsubseteq^* & \texttt{Manager} \sqcap \neg\texttt{TopManager} \\
\texttt{TopManager} & \sqsubseteq^* & \texttt{Manager} \\
\texttt{OrganisationalUnit} & \sqsubseteq^* & \texttt{Department} \sqcup \texttt{InterestGroup} \\
\texttt{Department} & \sqsubseteq^* & \texttt{OrganisationalUnit} \sqcap \neg\texttt{InterestGroup} \\
\texttt{InterestGroup} & \sqsubseteq^* & \texttt{OrganisationalUnit}
\end{array}$$

The snapshot cardinality constraints in $\Sigma_{\mathcal{S}}^{ex}$ give rise to the following formulas:

$$\texttt{Project} \sqsubseteq^* \exists^{\geq 1}[\texttt{act}]\texttt{Works-for} \sqcap \exists^{\geq 1}[\texttt{prj}]\texttt{Resp-for} \sqcap \exists^{=1}[\texttt{prj}]\texttt{Manages}$$
$$\texttt{TopManager} \sqsubseteq^* \exists^{=1}[\texttt{man}]\texttt{Manages}$$

The knowledge base Σ^{ex} introduces the binary relations `Works-for` between employees and projects, `Manages` between managers and projects, and `Resp-for` between project and organisational unit. Employees have exactly one pay slip number and one salary each, which are represented as binary relations with an integer domain. Furthermore, employees have exactly one name which is a string. Projects have exactly one project code which is a string. It is stated that managers are employees, and are partitioned into area managers and top managers, while organisational units are partitioned into departments and interest groups. 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 relations `Works-for` and `Resp-for`, and exactly once in the relation `Manages`.

5.2 Explicit temporal constructs

Explicit temporal support in $\mathcal{ER}_{VT}$ is obtained by further constraining the implicitly temporal constructs. In the following, the $\mathcal{DLR}_{\mathcal{US}}$ formulas used to encode the explicit temporal constructs are explained by referring to the example schema $\Sigma_{\mathcal{S}}^{ex}$.

Temporal entities and relations

Both entity and relationship instances in a temporal setting have an existence time associated with them. Formally, the function `Lifespan` maps pairs of objects and entity expressions (or n-tuples of objects and relationship expressions) to $2^{\mathcal{T}_p}$, i.e., it associates a *lifespan* to entities and relationship instances in the following way:

$$\texttt{Lifespan}(e\,, E) = \{t \in \mathcal{T}_p \mid e \in E^{\mathcal{B}(t)}\}$$
$$\texttt{Lifespan}(\langle e_1, \ldots, e_n\rangle,\ R) = \{t \in \mathcal{T}_p \mid \langle e_1, \ldots, e_n\rangle \in R^{\mathcal{B}(t)}\}$$

$\mathcal{DLR}_{\mathcal{US}}$-formulas can express *timestamps* by enforcing either that entities (relationships) cannot last forever—formulas **f11-12** for *temporary* entities and relationships, or that their extension never changes in time—formulas **f8-9** for *snapshot* entities relationships. In particular, temporary entities (relationships) are captured by saying that there must be a past or a future time point where the entity (relationship) does not hold. In other words, instances of temporary entities (relationships) always have a limited lifespan. On the other hand, snapshot entities (relationships) are captured by saying that whenever the entity (relationship) is true it is necessarily true in every past and every future time point, i.e, they never change along time. Snapshot entities and relationships are used to capture the semantics of *legacy* diagrams when included in a temporal model, thus enforcing the upward compatibility. Un-marked entities and relationships don't have a particular formula associated and they have temporally indeterminate instances. Let's now show how temporary or snapshot entities and relationships in $\Sigma_{\mathcal{S}}^{ex}$ are translated.

The explicitly snapshot entities in $\Sigma_{\mathcal{S}}^{ex}$ give rise to the following formulas:

$$\texttt{Employee} \sqsubseteq^* (\Box^+\texttt{Employee}) \sqcap (\Box^-\texttt{Employee})$$
$$\texttt{Department} \sqsubseteq^* (\Box^+\texttt{Department}) \sqcap (\Box^-\texttt{Department})$$

The explicitly snapshot relationship in $\Sigma_{\mathcal{S}}^{ex}$ gives rise to the following formula:

$$\texttt{Resp-for} \sqsubseteq^* (\Box^+\texttt{Resp-for}) \sqcap (\Box^-\texttt{Resp-for})$$

The explicitly temporary entity in $\Sigma_{\mathcal{S}}^{ex}$ gives rise to the following formula:

$$\texttt{Manager} \sqsubseteq^* (\Diamond^+\neg\texttt{Manager}) \sqcup (\Diamond^-\neg\texttt{Manager})$$

The explicitly temporary relationship in $\Sigma_{\mathcal{S}}^{ex}$ gives rise to the following formula:

$$\texttt{Works-for} \sqsubseteq^* (\Diamond^+\neg\texttt{Works-for}) \sqcup (\Diamond^-\neg\texttt{Works-for})$$

Temporal attributes

At different points in time, an entity may have different values for the same attribute. It is possible to associate a *validity time* to each attribute:

$$\texttt{ValidT}(\langle e, a\rangle, A) = \quad \{t \in \mathcal{T}_p \mid \langle e, a\rangle \in A^{\mathcal{B}(t)}\}$$

Snapshot attributes are captured in $\mathcal{DLR}_{\mathcal{US}}$ by constraining the attribute to have a global existence. Formula **f10** specifies that whenever an entity has a value for a snapshot attribute this value globally persists over time. On the other hand, temporary attributes have a limited validity time. Formula **f13** constraints a value of a temporary attribute to cease to be valid at a certain future or past time. Unmarked attributes admit both global and temporary values due to the implicit temporal semantics. Let's now show how the temporal properties of attributes in Σ_S^{ex} are translated.

The explicitly snapshot attribute in Σ_S^{ex} gives rise to the following formula:
$\texttt{Employee} \sqsubseteq^* \exists^{=1}[\texttt{From}](\Box^*\texttt{Name})$

The explicitly temporary attribute in Σ_S^{ex} gives rise to the following formula:
$\texttt{Employee} \sqsubseteq^* \exists^{=1}[\texttt{From}](\texttt{Salary} \sqcap (\Diamond^+\neg\texttt{Salary} \sqcup \Diamond^-\neg\texttt{Salary}))$

¿From the above formulas it is clear that the timestamp for an entity is independent from the timestamp of its attributes respecting the orthogonality principle. When considering the interaction between the temporal behaviour of an attribute and that of the owner entity, it is consistent in $\mathcal{DLR}_{\mathcal{US}}$ to have both snapshot attributes of a temporary entity, and temporary attributes of a snapshot entity. In the former case, the $\mathcal{DLR}_{\mathcal{US}}$ semantics says that during the (limited) lifespan of an entity the value of a snapshot attribute never changes. In the latter one, the meaning is that each instance always belongs to the snapshot entity, but the value of the temporary attribute may change during its existence. In our running example, where `Employee` is a snapshot entity, `Salary` is modelled as a temporary attribute—i.e., the salary of an employee will change, while its `Name`—represented as a snapshot attribute—persists over time. Furthermore, the temporal behaviour of an attribute is only specified locally to an entity, i.e., the same attribute associated to two different entities can have a different temporal behaviour—e.g., the phone number of a department is not supposed to change over time and should be modelled as a snapshot attribute, while the phone number of an employee can change and should be modelled as a temporary attribute.

Key constraints in Σ_S^{ex} give rise to the following formulas:

$$\begin{aligned}
\texttt{Employee} &\sqsubseteq^* \exists^{=1}[\texttt{From}](\Box^*\texttt{PaySlipNumber})\\
\top &\sqsubseteq^* \exists^{\leq 1}[\texttt{To}](\texttt{PaySlipNumber} \sqcap \texttt{From/2} : \texttt{Employee})\\
\texttt{Project} &\sqsubseteq^* \exists^{=1}[\texttt{From}](\Box^*\texttt{ProjectCode})\\
\top &\sqsubseteq^* \exists^{\leq 1}[\texttt{To}](\texttt{ProjectCode} \sqcap \texttt{From/2} : \texttt{Project})
\end{aligned}$$

Both `PaySlipNumber` and `ProjectCode` are modelled as snapshot (key) attributes that uniquely identifies an employee or a project, respectively. Notice that in our approach only single-attribute keys are captured. The case of full key dependencies in the a-temporal $\mathcal{DLR}$ language as been recently solved [Calvanese *et al.*, 2000] and its temporal extension has to be investigated.

As an advanced feature in $\mathcal{ER}_{VT}$ we mentioned the possibility to capture composite attributes. Indeed, let us assume that the entity E has the attribute A with components $A_1, \ldots, A_p$ then, instead of formula **f4**, the following $\mathcal{DLR}$ formula holds:

$$\begin{aligned}
E \sqsubseteq^* \exists^{=1}[\texttt{From}]A \sqcap \exists^{=1}[\texttt{From}](A \sqcap \texttt{To/2} : (&\exists^{=1}[\texttt{From}]A_1 \sqcap \exists^{=1}[\texttt{From}](A_1 \sqcap \texttt{To/2} : D_1) \sqcap \ldots \sqcap\\
&\exists^{=1}[\texttt{From}]A_p \sqcap \exists^{=1}[\texttt{From}](A_p \sqcap \texttt{To/2} : D_p)))
\end{aligned}$$

saying that the value of the attribute A participates, in its turn, to the attributes $A_1, \ldots, A_p$. Furthermore, $\mathcal{DLR}_{\mathcal{US}}$ can associate a different temporal behaviour to both the components and the owner attribute (in

the line of MADS). This is illustrated by showing how $\mathcal{DLR}_{\mathcal{US}}$ can capture the three example diagrams of Figure 2—for simplicity we consider only the Home-Phone component. Notice that the following formulas do not substitute the above formula for composite attributes but they have to be intended as additional constraints.

Case 1 gives rise to the following formula:

$$\texttt{Employee} \sqsubseteq^* \exists^{=1}[\texttt{From}](\texttt{Address} \sqcap (\Diamond^- \neg\texttt{Address} \sqcup \Diamond^+ \neg\texttt{Address}) \sqcap \texttt{To/2} : \exists^{=1}[\texttt{From}]\Box^*\texttt{Home-Phone})$$

Case 2 gives rise to the following formula:

$$\texttt{Employee} \sqsubseteq^* \exists^{=1}[\texttt{From}](\Box^*\texttt{Address} \sqcap \texttt{To/2} : \exists^{=1}[\texttt{From}](\texttt{Home-Phone} \sqcap (\Diamond^- \neg\texttt{Home-Phone} \sqcup \Diamond^+ \neg\texttt{Home-Phone})))$$

Case 3 gives rise to the following formula:

$$\texttt{Employee} \sqsubseteq^* \exists^{=1}[\texttt{From}](\texttt{Address} \sqcap (\Diamond^- \neg\texttt{Address} \sqcup \Diamond^+ \neg\texttt{Address}) \sqcap \texttt{To/2} : \exists^{=1}[\texttt{From}](\texttt{Home-Phone} \sqcap (\Diamond^- \neg\texttt{Home-Phone} \sqcup \Diamond^+ \neg\texttt{Home-Phone})))$$

Finally, we show how *multiple-valued* attributes are captured in $\mathcal{ER}_{VT}$. In case of multiple-valued attributes the formula **f4** has to be modified in the following way:

(f4') For each entity $E \in \mathcal{E}_{\mathcal{S}}$ such that $\text{ATT}_{\mathcal{S}}(E) = \langle A_1 : D_1, \ldots, A_h : D_h, A_{h+1} : D_{h+1}, \ldots, A_{h+p} : D_{h+p} \rangle$, where $A_i, \ldots, A_h$ are single-valued attributes and $A_{h+1}, \ldots, A_{h+p}$ are multiple-valued attributes, then the following formula holds:

$$\begin{aligned}\Phi(E) \sqsubseteq^* & \exists^{=1}[\texttt{From}]\Phi(A_1) \sqcap \exists^{=1}[\texttt{From}](\Phi(A_1) \sqcap \texttt{To/2} : \Phi(D_1)) \sqcap \ldots \sqcap \\ & \exists^{=1}[\texttt{From}]\Phi(A_h) \sqcap \exists^{=1}[\texttt{From}](\Phi(A_h) \sqcap \texttt{To/2} : \Phi(D_h)) \sqcap \ldots \sqcap \\ & \exists^{\geq 1}[\texttt{From}](\Phi(A_{h+1}) \sqcap \texttt{To/2} : \Phi(D_{h+1})) \sqcap \neg\exists^{\geq 1}[\texttt{From}](\Phi(A_{h+1}) \sqcap \texttt{To/2} : \neg\Phi(D_{h+1})) \\ & \sqcap \ldots \sqcap \\ & \exists^{\geq 1}[\texttt{From}](\Phi(A_{h+p}) \sqcap \texttt{To/2} : \Phi(D_{h+p})) \sqcap \neg\exists^{\geq 1}[\texttt{From}](\Phi(A_{h+p}) \sqcap \texttt{To/2} : \neg\Phi(D_{h+p}))\end{aligned}$$

For each multiple-valued attribute, the first conjunct says that there must be at least one value constrained to belong to a given basic domain while the second conjunct is essentially an universal quantification constraining all the possible values to belong to the given basic domain.

Temporal cardinalities

Cardinality constraints limit the participation of entities in relationships. In a temporal setting, we can distinguish between *snapshot participation constraints* (true at each point in time) and *lifespan participation constraints* [Tauzovich, 1991; Gregersen and Jensen, 1998; Spaccapietra *et al.*, 1998] (evaluated during the entire existence of the entity). While the cardinality construct in $\mathcal{DLR}$ is enough to capture snapshot participation constraints, $\mathcal{DLR}_{\mathcal{US}}$ is needed to capture the lifespan participation constraints. Let's now show how lifespan cardinality constraints in $\Sigma_{\mathcal{S}}^{ex}$ are translated.

The lifespan cardinality constraints in $\Sigma_{\mathcal{S}}^{ex}$ give rise to the following formula:

$$\texttt{TopManager} \sqsubseteq^* \exists^{\geq 1}[\texttt{man}](\Diamond^*\texttt{Manages}) \sqcap \exists^{\leq 5}[\texttt{man}](\Diamond^*\texttt{Manages})$$

i.e., a top manager should manage at least 1 and at most 5 different projects in his entire existence as a top manager. $\mathcal{DLR}_{\mathcal{US}}$ can also enforce a form of lifespan cardinality constraint with respect to the past or to the future only:

$E \sqsubseteq^* \exists^{\leqslant n}[U_i](\Diamond^- R)$ (*Past lifespan participation*)
$E \sqsubseteq^* \exists^{\leqslant n}[U_i](\Diamond^+ R)$ (*Future lifespan participation*)

Obviously, since for snapshot relationships the set of instances does not change in time there is no difference between snapshot and lifespan participation constraints.

5.3 Dynamic Temporal Features

$\mathcal{ER}_{VT}$ is able to capture both dynamic relationships between the entities of the schema, and a simplified form of schema evolution. Here we describe how $\mathcal{DLR}_{\mathcal{US}}$ can formalise those notions.

Dynamic entities

$\mathcal{DLR}_{\mathcal{US}}$ can model *transition* relationships between entities whose instances may migrate from one entity to the other; this is called *object migration* from a source to a target entity. In fact, in the temporal conceptual modelling literature, two notions of object migration are considered[Gupta and Hall, 1991; 1992; Spaccapietra *et al.*, 1998]: *evolution*, when an object ceases to be an instance of a source entity, and *extension*, when an object continues to belong to the source. Let E_s be the source entity and E_t be the target one; the case of dynamic extension is captured by the following formula:

$$E_s \sqsubseteq^* \Diamond^+ E_t \qquad \text{(\textit{Dynamic extension})}$$

Thus, every instance of the source entity must be an instance of the target at some moment in the future. To enforce the dynamic evolution the dynamic extension formula has to be restricted:

$$E_s \sqsubseteq^* \Diamond^+ (E_t \sqcap \neg E_s) \qquad \text{(\textit{Dynamic evolution})}$$

Specifying that when the target is reached the object doesn't belong anymore to the source. An interesting consequence of dynamic evolution is that the source is necessarily a temporary entity. The example diagrams of Figure 3 give rise to the following formulas:

$$\begin{aligned} \texttt{Student} &\sqsubseteq^* \Diamond^+ \texttt{Employee} \\ \texttt{ProbationaryEmployee} &\sqsubseteq^* \Diamond^+ (\texttt{ConfirmedEmployee} \sqcap \neg \texttt{ProbationaryEmployee}) \end{aligned}$$

The expressive power of $\mathcal{DLR}_{\mathcal{US}}$ can enforce more complex forms of dynamic behaviours. The *uni-directional evolution* is represented by adding a further constraint to the dynamic evolution:

$$E_s \sqsubseteq^* \Diamond^+ (E_t \sqcap \neg E_s \sqcap \Box^+ \neg E_s) \qquad \text{(\textit{Uni-directional evolution})}$$

i.e., when the target (E_t) is reached, the object cannot be involved a second time in the source (E_s). For example, there is a uni-directional evolution between being alive and being dead. It is also possible to constraint all the instances of the target to be involved in the migration process:

$$E_t \sqsubseteq^* \Diamond^- E_s \qquad \text{(\textit{Total Dynamic})}$$

i.e., every instance of the target entity was an instance of the source at some moment in the past. Adding this axiom to the previous ones it is possible to model: total dynamic extensions, total dynamic evolutions and total uni-directional evolutions. $\mathcal{DLR}_{\mathcal{US}}$ can also enforce the absence of an evolution between two entities:

$$E_s \sqsubseteq^* \Box^* \neg E_t \qquad \text{(\textit{Incompatible entities})}$$

A typical example of incompatible entities can be male and female. *Global consistency* restricts the dynamic behaviour of an entity in such a way that it is always populated by some instance at each moment of time:

$$\Box^* \neg (E \sqsubseteq \bot) \qquad \text{(\textit{Global consistency})}$$

i.e., at any moment of time, the entity E denotes a non empty set[7]. For example, it would be sensible to require that at any moment of time the company has at least one employee:

[7]Note that *global consistency* constraints all the temporal models to assign to the entity a non empty extension at all times, thus it is a constraint that selects exactly those temporal models where the entity is *globally satisfiable*.

$\Box^*\neg(\texttt{Employee} \sqsubseteq \bot)$.

Global consistency is an example of a so called *safety* constraint [Pnueli, 1986] which intuitively says that "*nothing bad ever happens*". On the other hand, *liveness* constraints [Pnueli, 1986] saying that "*something good will happen*" can be expressed by existential temporal formulas: $\Diamond^*\neg(E \sqsubseteq \bot)$, that constraint a temporal model to satisfy (not *globally*) the entity.

$\mathcal{DLR}_{\mathcal{US}}$ is also able to capture *generation* relationships where an instance (set of instances) from a source entity is (are) transformed in an instance of the target entity, and all the instances of the source are consumed in the transformation process. Let R be a generation relationship between the source entity E_s and the target entity E_t, then the following formulas hold:

$$\left.\begin{array}{rcl} R & \sqsubseteq^* & \texttt{source} : (\ominus E_s \sqcap \neg E_s \sqcap \Box^+\neg E_s) \sqcap \texttt{target} : (E_t \sqcap \Box^-\neg E_t) \\ E_s & \sqsubseteq^* & \exists^{\geq 1}[\texttt{source}]\Diamond^+ R \end{array}\right\} \text{(\textit{Generation relationship})}$$

i.e., a generation relationship generates new instances in the target entity starting from instances that where in the source entity at the previous time. The entities from the source are consumed—they do not belong anymore to the source, while the generated entities are new in the target—they where not instances of the target in any past time. Furthermore, all the instances of the source will be involved in the generation relationship. All the instances of the target can be constrained to be just the generated one:

$$E_t \sqsubseteq^* \exists^{\geq 1}[\texttt{target}]R \qquad \text{(\textit{Total transformation})}$$

The example diagram in Figure 3 gives rise to the following formulas:

$$\begin{array}{rcl} \texttt{Generate} & \sqsubseteq^* & \texttt{source} : (\ominus \texttt{Orange} \sqcap \neg\texttt{Orange} \sqcap \Box^+\neg\texttt{Orange}) \sqcap \texttt{target} : (\texttt{Juice} \sqcap \Box^-\neg\texttt{Juice}) \\ \texttt{Orange} & \sqsubseteq^* & \exists^{\geq 1}[\texttt{source}]\Diamond^+\texttt{Generate} \\ \texttt{Juice} & \sqsubseteq^* & \exists^{\geq 1}[\texttt{target}]\texttt{Generate} \end{array}$$

Schema evolution

The problem of schema evolution arises in the context of long-lived database applications, where stored data are considered worth surviving changes in the database schema [Roddick, 1995; Franconi *et al.*, 2000; Peters and Özsu, 1997]. According to a widely accepted terminology [Jensen *et al.*, 1998], a database supports *schema evolution* if it permits modifi cations of the schema without the loss of extant data. We consider here a simplifi ed case of conceptual schema evolution, which can be called a *monotonic* approach. This allows only for changes such that the resulting conceptual schema is compatible with the previous *global* constraints.

By using complex combinations of $\mathcal{DLR}_{\mathcal{US}}$-formulas, it is possible to express *conditional monotonic changes*. Let Γ and ϕ_p be atomic formulas, or possibly a Boolean combination of atomic formulas, which introduce, respectively, a new schema portion and a condition to be checked. The formula

$$\Box^*(\neg\phi_p \vee \Box^+\Gamma) \qquad \text{(\textit{Monotonic change})}$$

states that as soon as the property ϕ_p becomes true for the data, the conceptual schema will include the additional Γ constraint. A simple example is:

$$\begin{array}{l} \phi_{\mathtt{p}} \equiv (\texttt{InterestGroup} \sqsubseteq \bot) \\ \Gamma \equiv (\exists^{\geq 1}[\texttt{amount}](\texttt{Salary} \sqcap \texttt{payee}/2 : \texttt{TopManager}) \sqsubseteq \texttt{LowAmount}) \end{array}$$

meaning that as soon as the organisation does not include interest groups anymore, the salary of top managers should be in the "low amount" class.

Other subtler cases of schema evolution can be also expressed as $\mathcal{DLR}_{\mathcal{US}}$ formulas. Here we show a variant of *temporal existence dependency*: as soon as a condition becomes true, a particular entity starts

its existence, in the sense that before that moment it was necessarily empty. For example, the annual budget of a company ceases to be necessarily an empty entity only after every single project budget has been accepted by the administration:

$\Box^*((\texttt{CompanyBudget} \sqsubseteq \bot)\, \mathcal{U}\, (\texttt{ProjectBudget} \sqsubseteq \texttt{AcceptedBudget}))$

As a consequence of this formula, the entity `CompanyBudget` is a case of not globally satisfiable entity.

5.4 Extending $\mathcal{ER}_{VT}$

The correspondence established between $\mathcal{ER}_{VT}$ schemas and $\mathcal{DLR}_{\mathcal{US}}$ knowledge bases allows us to use the greater expressive power of $\mathcal{DLR}_{\mathcal{US}}$ to represent complex dependencies between elements of the schema. Using formulas in $\mathcal{DLR}_{\mathcal{US}}$ we can express complex temporal integrity constraints taking them in full account when reasoning over $\mathcal{ER}_{VT}$ schemas. For example, we can add the following formulas to our running knowledge base Σ^{ex}:

$\texttt{Employee} \sqcap \neg(\exists^{\geq 1}[\texttt{emp}]\texttt{Works-for}) \sqsubseteq^* \texttt{Manager}$
$\texttt{Manager} \sqsubseteq^* \neg(\exists^{\geq 1}[\texttt{emp}]\texttt{Works-for}) \sqcap (\texttt{Qualified}\ \mathcal{S}\ (\texttt{Employee} \sqcap \neg\texttt{Manager}))$

saying that 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 Σ^{ex} 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:

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

Note also that if we add to Σ^{ex} 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 Σ^{ex}, i.e., they are not satisfiable relative to Σ^{ex}.

6 Temporal queries

Another important reasoning task is known as the problem of query containment (see, e.g.,[Chomicki and Toman, 1998; Chomicki, 1994; Abiteboul *et al.*, 1996] 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 Σ is an expression of the form

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

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

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

P_j^i are $\mathcal{DLR}_{\mathcal{US}}$ entity or relation expressions, $\overrightarrow{x}^i_j$, $\overrightarrow{y}^i_j$, and $\overrightarrow{c}^i_j$ 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 Σ. This approach is similar to that of [Calvanese *et al.*, 1998a], where atoms in a query can be constrained by means of schema formulas. Furthermore, query expressions do not directly manipulate explicit temporal attributes, but time is implicit in each query expression. Indeed, the temporal dimension is handled by means of the temporal modal operators in $\mathcal{DLR}_{\mathcal{US}}$. In this perspective, the query language is in strict relation with the First-Order Temporal Logic over *since* and *until*, $L^{\{\textbf{since, until}\}}$, used in [Chomicki and Toman, 1998] for querying temporal databases.

The semantics of queries is based on the snapshot representation of a temporal database, and defined as follows. Given a temporal schema Σ, let $\mathcal{I}$ be a temporal model, and t be a time point in $\mathcal{T}$ such that $\mathcal{I}$ satisfies Σ 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 Σ, in the model $\mathcal{I}$ that satisfies Σ at moment t, and the *answer* to the query $\mathcal{Q}$, are respectively the sets:

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

We obtain a so called *sequenced semantics* [Böhlen *et al.*, 2000] for queries, which is based on the view of a database as a time-indexed collection of snapshots. The query language is also *Snapshot-Reducible* [Böhlen *et al.*, 2000] in the sense that non-temporal queries—i.e., queries without any temporal connective—are still valid queries, and are interpreted using the sequenced semantics. Our language allows also for *Upward Compatible* queries [Böhlen *et al.*, 2000]. Intuitively, a non-temporal query is upward compatible if the answer set on a temporal database is the same as the answer set on an associated non-temporal database. We obtain this behaviour by post-fixing a query with "@now" that does a temporal slice of the temporal database at the current time:

$$\mathsf{ans}(\mathcal{Q}@\mathsf{now}, \mathcal{I}) = \{\langle t, \overrightarrow{o}\rangle \in \mathcal{T} \times (\Delta^{\mathcal{I}})^n \mid t = \mathsf{now} \wedge \overrightarrow{o} \in \mathsf{eval}(\mathcal{Q}, \mathcal{I}(\mathsf{now}))\}$$

Example 6.1. We now show the expressivity of the query language by formulating some of the temporal queries used in [Chomicki and Toman, 1998].

1. "*Find all people who have worked for only one project*"
   $\mathcal{Q}(\mathtt{x}) \texttt{ :- } (\exists^{=1}[\mathtt{emp}](\Diamond^{*}\mathtt{Works\text{-}for}))(\mathtt{x})$
2. "*Find all managers whose terminal project has code* prj342"
   $\mathcal{Q}(\mathtt{x}) \texttt{ :- } \mathtt{Manager}(\mathtt{x}) \wedge \mathtt{Manages}(\mathtt{x}, \mathsf{prj342}) \wedge (\Box^{+}\neg\mathtt{Manages})(\mathtt{x}, \mathtt{y})$
3. "*Find all project-hoppers—people who never spent more than two consecutive years in a project*"
   $\mathcal{Q}(\mathtt{x}) \texttt{ :- } (\Box^{*}\neg\exists^{\geq 1}[\mathtt{emp}](\mathtt{Works\text{-}for} \sqcap \oplus \mathtt{Works\text{-}for} \sqcap \oplus \oplus \mathtt{Works\text{-}for}))(\mathtt{x})$
4. "*Find all people who did not work between two projects*"
   $\mathcal{Q}(\mathtt{x}) \texttt{ :- } (\Diamond^{-}\exists^{\geq 1}[\mathtt{emp}]\mathtt{Works\text{-}for})(\mathtt{x}) \wedge (\neg\exists^{\geq 1}[\mathtt{emp}]\mathtt{Works\text{-}for})(\mathtt{x}) \wedge (\Diamond^{+}\exists^{\geq 1}[\mathtt{emp}]\mathtt{Works\text{-}for})(\mathtt{x})$

We now formalise the notion of query containment. Given two queries (of the same arity) $\mathcal{Q}_1$ and $\mathcal{Q}_2$ over Σ, we say that $\mathcal{Q}_1$ is *contained* in $\mathcal{Q}_2$ under the constraints Σ, and write $\Sigma \models \mathcal{Q}_1 \subseteq \mathcal{Q}_2$, if, for every temporal model $\mathcal{I}$ of Σ we have $\mathsf{ans}(\mathcal{Q}_1, \mathcal{I}) \subseteq \mathsf{ans}(\mathcal{Q}_2, \mathcal{I})$. Note that the *query satisfiability problem*—given a query $\mathcal{Q}$ over a schema Σ, to determine whether there are $\mathcal{I}$ and t such that $\mathcal{I}, t \models \Sigma$, and $\mathsf{eval}(\mathcal{Q}, \mathcal{I}(t)) \neq \emptyset$—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 $\mathcal{Q}$.

Example 6.2. We now consider the problem of query containment under constraints, where the constraints are expressed by the $\mathcal{ER}_{VT}$ schema $\Sigma_{\mathcal{S}}^{ex}$ as illustrated in Figure 4. Consider the following query: "*Find all people that will be manager that work for a project, and all departments responsible for the project*":

$$\mathcal{Q}_1(\mathtt{x},\mathtt{y}) \ \mathtt{:\text{-}Works\text{-}for}(\mathtt{x},\mathtt{z}) \wedge \mathtt{Resp\text{-}for}(\mathtt{z},\mathtt{y}) \wedge \mathtt{Department}(\mathtt{y}) \wedge (\Diamond^{+}\mathtt{Manager})(\mathtt{x})$$

Given now the following query:

$$\mathcal{Q}_2(\mathtt{x},\mathtt{y}) \ \mathtt{:\text{-}Works\text{-}for}(\mathtt{x},\mathtt{z}) \wedge \mathtt{Resp\text{-}for}(\mathtt{z},\mathtt{y}) \wedge \neg\mathtt{InterestGroup}(\mathtt{y}) \wedge (\Diamond^{+}(\mathtt{AreaManager} \sqcup \mathtt{TopManager}))(\mathtt{x})$$

it is not hard to see that these two queries are equivalent under the constraints in Σ^{ex} (the $\mathcal{DLR}_{\mathcal{US}}$ translation of the $\mathcal{ER}_{VT}$ schema $\Sigma_{\mathcal{S}}^{ex}$), i.e.,

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

7 Decidability and complexity

In this section we analyse 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 7.1. *The global satisfiability problem for $\mathcal{DLR}_{\mathcal{US}}$ conceptual schemas containing only atomic formulas is undecidable.*

Remark 7.2. 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 proof (see Appendix) is based on the ability to simulate the $\mathbb{Z} \times \mathbb{N}$-grid in $\mathcal{DLR}_{\mathcal{US}}$; this is possible by temporalising binary relations (see the first two formulas in the proof). 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 [Wolter and Zakharyaschev, 1999a]. 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 7.3. *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.*

The language used in (1) is enough to capture most of the modelling constructs discussed in Section 4 (in fact, in the case of (global) logical implication for atomic formulas, there is no difference between $\sqsubseteq$ and $\sqsubseteq^{*}$), with the exception of (a) global entity consistency, (b) schema evolution constraints, (c) snapshot relations and attributes, and (d) temporal cardinalities. Full $\mathcal{DLR}_{\mathcal{US}}^{-}$ used in (2) is able to express (a) and (b). However, (c) and (d) require temporalised relations which, by Theorem 7.1, lead to undecidability.

8 Conclusion

This work introduces the temporal description logic $\mathcal{DLR}_{\mathcal{US}}$ and illustrates its expressive power by providing for the first time a systematic formalisation of various temporal entity-relationship conceptual data models appeared in the literature. 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.

Tight complexity results were proved. In particular, reasoning in the full logic $\mathcal{DLR}_{\mathcal{US}}$ was shown to be undecidable, while decidability was obtained using a still expressive fragment, $\mathcal{DLR}^{-}_{\mathcal{US}}$. We have shown 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 constraints with arbitrary $\mathcal{DLR}^{-}_{\mathcal{US}}$ formulas is decidable in 2EXPTIME with an EXPSPACE lower bound. The latter result is the first decidability result we are aware of on containment of temporal conjunctive queries under expressive constraints.

References

[Abiteboul *et al.*, 1996] 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.

[Artale and Franconi, 1999a] A. Artale and E. Franconi. Reasoning with enhanced temporal entity-relationship models. In *Proc. of the International Workshop on Spatio-Temporal Data Models and Languages*. IEEE Computer Society Press, August 1999. Also in Proc. of the 6^{th} International Workshop on Knowledge Representation meets Databases (KRDB'99), and in Proc. of the 1999 International Workshop on Description Logics (DL'99).

[Artale and Franconi, 1999b] 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.

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

[Böhlen *et al.*, 2000] M. Böhlen, C. Jensen, and R. Snodgrass. Temporal statement modifiers. *ACM Transactions On Database Systems*, 2000.

[Calvanese *et al.*, 1998a] 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.

[Calvanese *et al.*, 1998b] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Information integration: Conceptual modeling and reasoning support. In *Proc. of the 6th Int. Conf. on Cooperative Information Systems (CoopIS'98)*, pages 280–291, 1998.

[Calvanese *et al.*, 1998c] D. Calvanese, M. Lenzerini, and D. Nardi. Description logics for conceptual data modeling. In Chomicki and Saake [1998].

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

[Calvanese *et al.*, 2000] D. Calvanese, G. De Giacomo, and M. Lenzerini. Keys for free in description logics. In *Proc. of the 2000 International Workshop on Description Logics (DL'2000)*, 2000.

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

[Chomicki and Toman, 1998] J. Chomicki and D. Toman. Temporal logic in information systems. In Chomicki and Saake [1998], chapter 1.

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

[Elmasri and Navathe, 1994] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems*. Benjamin/Cummings, 2nd edition, 1994.

[Franconi and Ng, 2000] 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.

[Franconi *et al.*, 2000] E. Franconi, F. Grandi, and F. Mandreoli. A semantic approach for schema evolution and versioning in object-oriented databases. In *Proc. of the 1st International Conf. on Computational Logic (CL'2000), DOOD stream*. Springer-Verlag, July 2000.

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

[Gregersen and Jensen, 1998] H. Gregersen and J.S. Jensen. Conceptual modeling of time-varying information. Technical Report TimeCenter TR-35, Aalborg University, Denmark, 1998.

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

[Gupta and Hall, 1991] R. Gupta and G. Hall. Modeling transition. In *Proc. of ICDE'91*, pages 540–549, 1991.

[Gupta and Hall, 1992] R. Gupta and G. Hall. An abstraction mechanism for modeling generation. In *Proc. of ICDE'92*, pages 650–658, 1992.

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

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

[Horrocks *et al.*, 2000] 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.

[Horrocks, 1999] I. Horrocks. FaCT and iFaCT. In *Proceedings of the International Workshop on Description Logics (DL'99)*, pages 133–135, 1999.

[Jarke *et al.*, 2000] 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.

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

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

[Jensen *et al.*, 1998] C. S. Jensen, J. Clifford, S. K. Gadia, P. Hayes, and S. Jajodia et al. The Consensus Glossary of Temporal Database Concepts. In O. Etzion, S. Jajodia, and S. Sripada, editors, *Temporal Databases - Research and Practice*, pages 367–405. Springer-Verlag, 1998.

[Levy and Rousset, 1998] A. Y. Levy and M-C. Rousset. Combining horn rules and description logics in CARIN. *Artificial Intelligence*, 104(1-2):165–209, 1998.

[Peters and Özsu, 1997] R. J. Peters and M. T. Özsu. An Axiomatic Model of Dynamic Schema Evolution in Objectbase Systems. *ACM Transactions On Database Systems*, 22(1):75–114, 1997.

[Pnueli, 1986] A. Pnueli. Application of temporal logic to the specification and verification of reactive systems: A survey of current trends. In *Current Trends in Concurrency*, volume 224 of *LNCS*, pages 510–584. Springer-Verlag, 1986.

[Robinson, 1971] R. Robinson. Undecidability and non-periodicity for tilings of the plan. *Invent. Math.*, 12:177–209, 1971.

[Roddick, 1995] J. F. Roddick. A Survey of Schema Versioning Issues for Database Systems. *Information and Software Technology*, 37(7):383–393, 1995.

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

[Simons, 1987] Peter Simons. *Parts: A Study in Ontology*. Clarendon Press, Oxford, 1987.

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

[Snodgrass, 1987] R. T. Snodgrass. The temporal query language TQUEL. *ACM Transaction on Database Systems*, 12(2):247–298, 1987.

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

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

[Tauzovich, 1991] B. Tauzovich. Towards temporal extensions to the entity-relationship model. In *Proc. of the* 10^{th} *International Conference on the Entity-Relationship Approach (ER'91)*, pages 163–179, 1991.

[Theodoulidis *et al.*, 1991] C. Theodoulidis, P. Loucopoulos, and B. Wangler. A conceptual modelling formalism for temporal database applications. *Information Systems*, 16(3):401–416, 1991.

[Wolter and Zakharyaschev, 1998] 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.

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

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

[Wolter and Zakharyaschev, 1999c] F. Wolter and M. Zakharyaschev. Temporalizing description logics. In D. Gabbay and M. de Rijke, editors, *Frontiers of Combining Systems*. Studies Press-Wiley, 1999. Also in the Proceedings of *FroCoS'98*, Amsterdam, 1998.

Appendix: proofs

A A formalisation of $\mathcal{ER}_{VT}$

Proposition 2.5. In every $\mathcal{ER}_{VT}$ schema the following temporal properties hold:

1. Sub-entities of temporary entities are also temporary.
2. Sub-entities of snapshot entities, and super-entities of temporary or un-marked entities can be either snapshot, temporary or un-marked entities.
3. Super-entities of snapshot entities are also snapshot.
4. A schema is inconsistent if exactly one of a whole set of snapshot partitioning sub-entities is temporary.
5. Participants of snapshot relations are either snapshot or un-marked entities. They are snapshot when they participate at least once in the relationship.
6. Participants of temporary or un-marked relations can be either snapshot, temporary or un-marked entities.
7. A relationship is temporary if one of the participating entities is temporary.
8. The temporal behaviour for an entity is independent from that of its attributes.

Points $1-3$ are true also for relationships.

Proof We prove two cases, the rest easily follows.
(1) Let assume that $E_1 \text{ISA}_{\mathcal{S}} E_2$ and E_2 is a temporary entity. Then we prove that there cannot be an instance, e, of E_1 such that $\forall t \in \mathcal{T}.e \in E_1^{\mathcal{I}(t)}$. Indeed, if this is the case then for $t_0 \in \mathcal{T}, e \in E_1^{\mathcal{I}(t_0)}$ and, by the $\text{ISA}_{\mathcal{S}}$ statement, $e \in E_2^{\mathcal{I}(t_0)}$. Since E_2 is temporary $\exists t_1 \neq t_0.e \notin E_2^{\mathcal{I}(t_1)}$ which implies that $e \notin E_1^{\mathcal{I}(t_1)}$ that is a contradiction.

(5) We first prove that the entity cannot be temporary. By absurd, let R be a snapshot relationship and E_i be a participating temporary entity. Now, let $\langle e_1, \ldots, e_i, \ldots, e_n \rangle$ an instance of R, then, $\forall t \in \mathcal{T}.\langle e_1, \ldots, e_i, \ldots, e_n \rangle \in R^{\mathcal{I}(t)}$. This implies that $\forall t \in \mathcal{T}.e_i \in E_i^{\mathcal{I}(t)}$, which contradicts the assumption that E_i is a temporary entity—this prove the first part. If now E_i participates at least once in R, then, $\forall e_i \in E_i^{\mathcal{I}(t)}.\langle e_1, \ldots, e_i, \ldots, e_n \rangle \in R^{\mathcal{I}(t)}$, but $\langle e_1, \ldots, e_i, \ldots, e_n \rangle \in R^{\mathcal{I}(t)}$, for all $t \in \mathcal{T}$, and then also $e_i \in E_i^{\mathcal{I}(t)}$, for all $t \in \mathcal{T}$, i.e., E_i is a snapshot entity. ❑

B Encoding $\mathcal{ER}_{VT}$ in $\mathcal{DLR}_{\mathcal{US}}$

Proposition 4.3. Let $\Sigma_{\mathcal{S}} = (\mathcal{L}_{\mathcal{S}}, \text{REL}_{\mathcal{S}}, \text{ATT}_{\mathcal{S}}, \text{CARD}_{\mathcal{S}}, \text{CARD}^{L}_{\mathcal{S}}, \text{ISA}_{\mathcal{S}}, \text{DISJ}_{\mathcal{S}}, \text{COVER}_{\mathcal{S}}, \text{DISCOVER}_{\mathcal{S}}, \text{S}_{\mathcal{S}}, \text{T}_{\mathcal{S}}, \text{KEY}_{\mathcal{S}})$ be an $\mathcal{ER}_{VT}$ schema, then:

1. For each legal temporal database state $\mathcal{B}$ for $\Sigma_{\mathcal{S}}$ there is a temporal model $\mathcal{I}$ of $\Phi(\Sigma_{\mathcal{S}})$ such that for each symbol $X \in \mathcal{E}_{\mathcal{S}} \cup \mathcal{A}_{\mathcal{S}} \cup \mathcal{R}_{\mathcal{S}} \cup \mathcal{D}_{\mathcal{S}}$ then $\Phi(X)^{\mathcal{I}(t)} = X^{\mathcal{B}(t)}$, for each $t \in \mathcal{T}$.
2. For each temporal model $\mathcal{I} = \langle \mathcal{T}, \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}(t)} \rangle$ of $\Phi(\Sigma_{\mathcal{S}})$ there is a legal temporal database state $\mathcal{B}$ for $\Sigma_{\mathcal{S}}$, a set of basic domains *BD*, and a one-to-one partial function $\mathcal{B}_{\Delta} : \Delta^{\mathcal{I}} \rightarrow BD$—total on $\bigcup_{D_i \in \mathcal{D}_{\mathcal{S}}} \Phi(D_i)^{\mathcal{I}(t)}$—such that:

(a) For each symbol $X \in \mathcal{E}_\mathcal{S} \cup \mathcal{R}_\mathcal{S}$: $X^{\mathcal{B}(t)} = \Phi(X)^{\mathcal{I}(t)}$, for each $t \in \mathcal{T}$;

(b) For each $D_i \in \mathcal{D}_\mathcal{S}$: $D_i^{\mathcal{B}(t)} = \mathcal{B}_\Delta(\Phi(D_i)^{\mathcal{I}(t)})$, for each $t \in \mathcal{T}$;

(c) For each $A \in \mathcal{A}_\mathcal{S}$: $\langle d_1, d_2\rangle \in A^{\mathcal{I}(t)}$ iff $\langle d_1, \mathcal{B}_\Delta(d_2)\rangle \in A^{\mathcal{B}(t)}$, for each $t \in \mathcal{T}$.

Proof (1) Let $\mathcal{B} = (\mathcal{T}, \Delta^\mathcal{B} \cup \Delta_D^\mathcal{B}, \cdot^{\mathcal{B}(t)})$ be a legal temporal database state. We define a temporal model $\mathcal{I} = (\mathcal{T}, \Delta^\mathcal{I}, \cdot^{\mathcal{I}(t)})$ of $\Phi(\Sigma_\mathcal{S})$ as follows:

- $\Delta^\mathcal{I} = \Delta^\mathcal{B} \cup \Delta_D^\mathcal{B}$.
- For each symbol $X \in \mathcal{E}_\mathcal{S} \cup \mathcal{A}_\mathcal{S} \cup \mathcal{R}_\mathcal{S} \cup \mathcal{D}_\mathcal{S}$ then $\Phi(X)^{\mathcal{I}(t)} = X^{\mathcal{B}(t)}$, for each $t \in \mathcal{T}$.

We prove that such a model, $\mathcal{I}$, satisfies all the formulas (`f1-f18`) in $\Phi(\Sigma_\mathcal{S})$.

(f1) $\forall t \in \mathcal{T}, \forall D_i \in \mathcal{D}_\mathcal{S}.\Phi(D_i)^{\mathcal{I}(t)} = D_i^{\mathcal{B}(t)} \equiv \Delta_{D_i}^\mathcal{B}$, by definition of $\mathcal{I}$ and $\mathcal{B}$, i.e., $\forall t_1, t_2 \in \mathcal{T}.\Phi(D_i)^{\mathcal{I}(t_1)} = \Phi(D_i)^{\mathcal{I}(t_2)}$.

(f2) Let $R \in \mathcal{R}_\mathcal{S}$ be such that $\text{REL}_\mathcal{S}(R) = \langle U_1 : E_1, \ldots, U_k : E_k\rangle$, and let $r \in R^{\mathcal{B}(t)}$. Then, by definition of $\mathcal{I}$, $r \in \Phi(R)^{\mathcal{I}(t)}$. So, $r = \langle U_1 : e_1, \ldots, U_k : e_k\rangle \equiv \langle e_1, \ldots, e_k\rangle$, with $e_i \in E_i^{\mathcal{B}(t)} = \Phi(E_i)^{\mathcal{I}(t)}$, for $i \in \{1, \ldots, k\}$.

(f6.a) For each role symbol $U_i \in \mathcal{U}_\mathcal{S}$ between $R \in \mathcal{R}_\mathcal{S}$ and $E \in \mathcal{E}_\mathcal{S}$, if $m = \text{CMIN}_\mathcal{S}^L(E, R, U_i) \neq 0$ then $\forall e \in E^{\mathcal{B}(t)}.\text{CMIN}_\mathcal{S}^L(E, R, U_i) \leq \#\bigcup_{t' \in \mathcal{T}}\{r \in R^{\mathcal{B}(t')} \mid r[U_i] = e\}$. By definition of $\mathcal{I}$, $e \in \Phi(E)^{\mathcal{I}(t)}$, and the proof follows by considering that: $\#\{r \in (\Diamond^* R)^{\mathcal{I}(t)} \mid r[\Phi(U_i)] = e\} \equiv \#\bigcup_{t' \in \mathcal{T}}\{r \in R^{\mathcal{I}(t')} \mid r[\Phi(U_i)] = e\} \equiv \#\bigcup_{t' \in \mathcal{T}}\{r \in R^{\mathcal{B}(t')} \mid r[U_i] = e\}$.

(f10) For each entity $E \in \mathcal{E}_\mathcal{S}$ if $\langle E, A_i\rangle \in \text{S}_\mathcal{S}$ then $\forall e \in E^{\mathcal{B}(t)}.\langle e, a_i\rangle \in A_i^{\mathcal{B}(t)} \rightarrow \forall t' \in \mathcal{T}.\langle e, a_i\rangle \in A_i^{\mathcal{B}(t')}$. By definition of $\mathcal{I}$, $e \in \Phi(E)^{\mathcal{I}(t)}$ and $\forall t' \in \mathcal{T}.\langle e, a_i\rangle \in \Phi(A_i)^{\mathcal{I}(t')}$.

The proof of the remaining formulas is similar to the above cases.

(2) Let $\mathcal{I} = (\mathcal{T}, \Delta^\mathcal{I}, \cdot^{\mathcal{I}(t)})$ be a temporal model of $\Phi(\Sigma_\mathcal{S})$. We first define the set of active domains in $\Sigma_\mathcal{S}$, $\Delta_D^\mathcal{B}$, starting from $\Phi(D_i)^{\mathcal{I}(t)}$. By formula `f1`, the interpretation of $\Phi(D_i)$ is time-invariant, and, by formula `f14`, each $\Phi(D_i)$ denotes a set disjoint from all the other $\Phi(D_j)$, for $i \neq j$. Given a set of basic domains BD, and a one-to-one partial function $\mathcal{B}_\Delta : \Delta^\mathcal{I} \rightarrow BD$, we choose $\Delta_{D_i}^\mathcal{B} = \mathcal{B}_\Delta(\Phi(D_i)^{\mathcal{I}(t)})$, for some $t \in \mathcal{T}$, and $\Delta_D^\mathcal{B} = \bigcup_{D_i \in \mathcal{D}_\mathcal{S}} \Delta_{D_i}^\mathcal{B}$. We can now define the temporal database state $\mathcal{B} = (\mathcal{T}, \Delta^\mathcal{B} \cup \Delta_D^\mathcal{B}, \cdot^{\mathcal{B}(t)})$:

- $\Delta^\mathcal{B} = \Delta^\mathcal{I} \setminus \bigcup_{D_i \in \mathcal{D}_\mathcal{S}} \Phi(D_i)^{\mathcal{I}(t)}$, for some $t \in \mathcal{T}$.
- $\cdot^{\mathcal{B}(t)}$ verifies the conditions *(a-c)* stated in the theorem.

$\mathcal{B}$ can be proven to be a legal temporal database state by showing that $\mathcal{B}$ satisfies all the constraints of Definition 2.3.

(c1) Follows from formula `f7.a` and the fact that, by definition of $\mathcal{B}$, $\forall E \in \mathcal{E}_\mathcal{S}, \forall t \in \mathcal{T}.E^{\mathcal{B}(t)} = E^{\mathcal{I}(t)}$.

(c3) For each $E \in \mathcal{E}_\mathcal{S}$ if $\text{ATT}_\mathcal{S}(E) = \langle A_1 : D_1, \ldots, A_h : D_h\rangle$ then, by formula `f4`, $\forall e \in \Phi(E)^{\mathcal{I}(t)}, \forall i \in \{1, \ldots, h\}, \exists! d_i.\langle e, d_i\rangle \in \Phi(A_i)^{\mathcal{I}(t)} \wedge \forall d_i.(\langle e, d_i\rangle \in \Phi(A_i)^{\mathcal{I}(t)} \rightarrow d_i \in \Phi(D_i)^{\mathcal{I}(t)})$. By definition of $\mathcal{B}$ and the $\mathcal{B}_\Delta$ function, $e \in E^{\mathcal{B}(t)}$, $\mathcal{B}_\Delta(d_i) = a_i \in D_i^{\mathcal{B}(t)} = \Delta_{D_i}^\mathcal{B}$, and $\langle e, a_i\rangle \in A_i^{\mathcal{B}(t)}$.

(c6) For each role symbol $U_i \in \mathcal{U}_\mathcal{S}$ between $R \in \mathcal{R}_\mathcal{S}$ and $E \in \mathcal{E}_\mathcal{S}$, if $m = \text{CMIN}^L_\mathcal{S}(E, R, U_i) \neq 0$ then, by formula `f6.a`, $\forall e \in \Phi(E)^{\mathcal{I}(t)}.\#\{r \in (\Diamond^*\Phi(R))^{\mathcal{I}(t)} \mid r[\Phi(U_i)] = e\} \geq m$, i.e., by the semantics of $\Diamond^*\Phi(R)$, $\forall e \in \Phi(E)^{\mathcal{I}(t)}.\#\bigcup_{t' \in \mathcal{T}}\{r \in \Phi(R)^{\mathcal{I}(t')} \mid r[\Phi(U_i)] = e\} \geq m$. Now, by definition of $\mathcal{B}$, $e \in E^{\mathcal{B}(t)}$ and $r \in R^{\mathcal{B}(t')}$, the case where $\text{CMAX}^L_\mathcal{S}(E, R, U_i) \neq \text{N}$ is similar, then `c6` is true.

(c7) For each snapshot entity $E \in \mathcal{E}^S_\mathcal{S}$ formula `f8` is true: $\Phi(E) \doteq \Box^*\Phi(E)$. Then, $e \in E^{\mathcal{I}(t)}$ implies that $\forall t' \in \mathcal{T}.e \in E^{\mathcal{I}(t')}$. By definition of $\mathcal{B}$, constraint `c7` easily follows.

(c8) For each temporary entity $E \in \mathcal{E}^T_\mathcal{S}$ formula `f11` is true: $\Phi(E) \sqsubseteq^* (\Diamond^+\neg\Phi(E)) \sqcup (\Diamond^-\neg\Phi(E))$. Then, $e \in E^{\mathcal{I}(t)}$ implies that $\exists t_1 > t.e \notin E^{\mathcal{I}(t_1)} \vee \exists t_2 < t.e \notin E^{\mathcal{I}(t_2)}$. By definition of $\mathcal{B}$, constraint `c8` easily follows.

(c11) For each entity $E \in \mathcal{E}_\mathcal{S}$ if $\text{ATT}_\mathcal{S}(E) = \langle A_1 : D_1, \ldots, A_h : D_h\rangle$ and $\langle E, A_i\rangle \in \text{S}_\mathcal{S}$, then formula `f10` is true. Then, $\forall e \in E^{\mathcal{I}(t)}.\langle e, d_i\rangle \in A_i^{\mathcal{I}(t)} \rightarrow \forall t' \in \mathcal{T}.\langle e, d_i\rangle \in A_i^{\mathcal{B}(t')}$. By definition of $\mathcal{B}$ and the $\mathcal{B}_\Delta$ function, the constraint `c11` easily follows.

The proof of the remaining constraints is similar to the above cases. ∎

C Decidability and complexity

Theorem 7.1. The global satisfiability problem for $\mathcal{DLR}_{\mathcal{US}}$ conceptual schemas containing only atomic formulas is undecidable.

Proof The proof is by reduction of the well-known undecidable tiling problem[Robinson, 1971]: given a finite set of square tiles of fixed orientation and with coloured edges, decide whether it can tile the grid $\mathbb{Z} \times \mathbb{N}$. Suppose $\mathfrak{T} = \{T_1, \ldots, T_k\}$ is a set of tiles with colours *left*(T_i), *right*(T_i), *up*(T_i), and *down*(T_i). Consider the following schema Σ, where $D_1, \ldots, D_k$ are concepts and R is a binary relation:

- $R \doteq \Box^+R,\ R \doteq \Diamond^+R,\ \top \doteq \exists R.\top$,
- $D_i \sqsubseteq \neg D_j,\ \top \doteq D_1 \sqcup \cdots \sqcup D_k$, for $i \neq j$,
- $D_i \sqsubseteq \bigsqcup_{\textit{right}(T_i)=\textit{left}(T_j)} \forall R.D_j$, for $i \leq k$,
- $D_i \sqsubseteq \bigsqcup_{\textit{up}(T_i)=\textit{down}(T_j)} \oplus D_j$, for $i \leq k$.

(Here $\exists R.C = \exists^{\geq 1}[1](R \sqcap 2/2 : C)$, $\forall R.C = \neg\exists R.\neg C$.) It is readily checked that Σ is globally satisfiable iff $\mathfrak{T}$ tiles $\mathbb{Z} \times \mathbb{N}$. ∎

Theorem 7.3. 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 the remainder of the section we sketch a proof of this theorem. To make it more transparent, we confine ourselves to considering only the 'future fragment' $\mathcal{DLR}_\mathcal{U}$ of $\mathcal{DLR}^-_{\mathcal{US}}$. (From now on $\Box$ stands

for $\Box^+$ and $\bigcirc$ for $\oplus$.) The main technical tool in the proof is the method of quasimodels developed in [Wolter and Zakharyaschev, 1998; 1999b]. 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.

Suppose that Γ consists of a finite set $f(\Gamma)$ of $\mathcal{DLR}^-_{\mathcal{U}}$-formulas and a finite set $c(\Gamma)$ of concepts, $f(\Gamma)$ is closed under sub-formulas, $c(\Gamma)$ under subconcepts, both are closed under (single) negation, and each concept occurring in $f(\Gamma)$ belongs $c(\Gamma)$. A *concept type* for Γ is a subset t of $c(\Gamma)$ such that

- $C \sqcap D \in t$ iff $C, D \in t$, for all $C \sqcap D \in c(\Gamma)$;
- $\neg C \in t$ iff $C \notin t$, for all $C \in c(\Gamma)$.

A *formula type* for Γ is a subset Φ of $f(\Gamma)$ such that

- $\psi \wedge \chi \in \Phi$ iff $\psi, \chi \in \Phi$, for all $\psi \wedge \chi \in f(\Gamma)$;
- $\neg\psi \in \Phi$ iff $\psi \notin \Phi$, for all $\psi \in f(\Gamma)$.

A pair $\langle T, \Phi\rangle$, where T is a set of concept types and Φ a formula type for Γ, is called a *quasistate candidate* for Γ. We say that the quasistate candidate $\mathfrak{C} = \langle T, \Phi\rangle$ is a *quasistate* for Γ if the following (non-temporal) $\mathcal{DLR}$-formula $\alpha_{\mathfrak{C}}$

$$\Big(\bigsqcup_{t\in T} c(t) \doteq \top\Big) \wedge \bigwedge_{t\in T} \neg(c(t) \doteq \bot) \wedge \bigwedge \Phi$$

is satisfiable. Here $c(t)$ denotes the conjunction of all concepts in t, concepts of the form $C\,\mathcal{U}\,D$ are regarded as atomic concepts $A_{C\mathcal{U}D}$, and formulas of the form $\varphi\mathcal{U}\psi$ in Φ are regarded as atomic formulas $A_{\varphi\mathcal{U}\psi} = \top$.

Consider now a sequence of quasistates

$$Q = \langle Q(n) : n \in \mathbb{Z}\rangle\,,$$

where $Q(n) = \langle T_n, \Phi_n\rangle$. A *run* in Q is a sequence $r = \langle r(n) : n \in \mathbb{Z}\rangle$ such that

1. $r(n) \in T_n$ for every $n \in \mathbb{Z}$;
2. for every $C\,\mathcal{U}\,D \in c(\Gamma)$ and every $n \in \mathbb{Z}$, we have $C\,\mathcal{U}\,D \in r(n)$ iff there is $l > n$ such that $D \in r(l)$ and $C \in r(k)$ for all $k \in (n, l)$.

Finally, Q is called a *quasimodel for* Γ if the following conditions hold:

3. for every $n \in \mathbb{Z}$ and every $t \in T_n$ there is a run r in Q such that $r(n) = t$;
4. for every $\psi\,\mathcal{U}\,\chi \in f(\Gamma)$ and every $n \in \mathbb{Z}$, we have $\psi\,\mathcal{U}\,\chi \in \Phi_n$ iff there is $l > n$ such that $\chi \in \Phi_l$ and $\psi \in \Phi_k$ for all $k \in (n, l)$.

Given a $\mathcal{DLR}^-_{\mathcal{U}}$-formula φ, we denote by $cl(\varphi)$ the closure under (single) negation of the set of subformulas and subconcepts of φ.

Theorem C.1. *A $\mathcal{DLR}^-_{\mathcal{U}}$-formula φ is satisfiable iff there is a quasimodel for $cl(\varphi)$ such that $\varphi \in \Phi_0$.*

Proof Suppose φ is satisfied in a model $\mathcal{I}$ with domain Δ. For every $n \in \mathbb{Z}$, define $Q(n) = \langle T_n, \Phi_n\rangle$ by taking

$$T_n = \{t^n(x) : x \in \Delta\},\ \Phi_n = \{\psi \in cl(\varphi) : \mathcal{I}, n \models \psi\},$$

where $t^n(x) = \{C \in cl(\varphi) : x \in C^{\mathcal{I}(n)}\}$. It is not hard to see that $\langle Q(n) : n \in \mathbb{Z}\rangle$ is a quasimodel for φ. (Note that the sequence $\langle t^n(x) : n \in \mathbb{Z}\rangle$ is a run through $t^n(x)$, for every $n \in \mathbb{Z}$ and every $x \in \Delta$). To show the converse we require the following lemma.

Lemma C.2. *For any cardinal $\kappa \geq \aleph_0$ and any quasistate $\mathfrak{C}$ for φ, the formula $\alpha_{\mathfrak{C}}$ is satisfied in a (non-temporal) $\mathcal{DLR}$-model $\mathcal{J}$ in which $\left|[x]^{\mathcal{J}}\right| = \kappa$ for all x in the domain Δ of $\mathcal{J}$, where*

$$[x]^{\mathcal{J}} = \{y \in \Delta : \forall C \in cl(\varphi)(x \in C^{\mathcal{J}} \Leftrightarrow y \in C^{\mathcal{J}})\}.$$

Proof As $\mathcal{DLR}$ is a fragment of first-order logic, we have a countable $\mathcal{DLR}$-model $\mathcal{I}$ satisfying $\alpha_{\mathfrak{C}}$. Define $\mathcal{J}$ as the disjoint union of κ copies of $\mathcal{I}$; more precisely, let

$$\Delta^{\mathcal{J}} = \{\langle x, \xi\rangle : x \in \Delta^{\mathcal{I}}, \xi < \kappa\},$$

$$P_i^{\mathcal{J}} = \{\langle \langle x_0, \xi\rangle, \ldots, \langle x_n, \xi\rangle\rangle : \langle x_0, \ldots, x_n\rangle \in P_i^{\mathcal{I}}, \xi < \kappa\},$$

$$(\top_n)^{\mathcal{J}} = \{\langle \langle x_0, \xi\rangle, \ldots, \langle x_n, \xi\rangle\rangle : \langle x_0, \ldots, x_n\rangle \in (\top_n)^{\mathcal{I}}, \xi < \kappa\}.$$

It is not hard to see that $\mathcal{J}$ is as required. ❑

Suppose now that $\varphi \in \Phi_0$, for a quasimodel Q. Let κ be a cardinal exceeding the cardinality of the set Ω of all runs in Q and $\aleph_0$, and let

$$\Delta = \{\langle r, \xi\rangle : r \in \Omega, \xi < \kappa\}.$$

Note that $|\{\langle r, \xi\rangle \in \Delta : r(n) = t\}| = \kappa$, for every $n \in \mathbb{Z}$ and every $t \in T_n$. By Lemma C.2, for every $n \in \mathbb{Z}$ there is a $\mathcal{DLR}$-model $\mathcal{J}(n)$ with domain Δ satisfying $\alpha_{Q(n)}$ and such that $\{C \in cl(\varphi) : \langle r, \xi\rangle \in C^{\mathcal{J}(n)}\} = r(n)$, for all $r \in \Omega$ and $\xi < \kappa$. It is not hard to see that the temporal $\mathcal{DLR}$-model $\mathcal{I} = \left\langle \mathbb{Z}, \Delta, \cdot^{\mathcal{I}(n)}\right\rangle$ defined by taking $\mathcal{I}(n) = \mathcal{J}(n)$, for every $n \in \mathbb{Z}$, satisfies φ at moment 0. ❑

Thus, the satisfiability problem for $\mathcal{DLR}_{\mathcal{U}}^{-}$-formulas reduces to checking 'satisfiability' in quasimodels. Consider now a $\mathcal{DLR}_{\mathcal{U}}^{-}$-schema Σ and two queries

$$\mathcal{Q}_i(\vec{x}) \text{ :- } \bigvee_j \mathcal{Q}_j^i(\vec{x}, \vec{y_{ij}}, \vec{w_{ij}}), \quad i = 1, 2.$$

Denote by $cl(\Sigma, \mathcal{Q}_1, \mathcal{Q}_2)$ the closure under (single) negation of the set of all formulas and concepts occurring in Σ, $\mathcal{Q}_1$ and $\mathcal{Q}_2$. Given a formula or a concept χ, denote by $\overline{\chi}$ the result of replacing all subformulas (subconcepts) in χ of the form $\chi_1 \,\mathcal{U}\, \chi_2$ with $A_{\chi_1 \mathcal{U} \chi_2} = \top$ (respectively, $A_{\chi_1 \mathcal{U} \chi_2}$). Thus, $\overline{\chi}$ is a $\mathcal{DLR}$ formula or concept, and the $\overline{\mathcal{Q}_i}$ are non-temporal $\mathcal{DLR}$-queries.

Theorem C.3. *$\mathcal{Q}_1$ is not contained in $\mathcal{Q}_2$ relative to Σ iff there is a quasimodel Q for $cl(\Sigma, \mathcal{Q}_1, \mathcal{Q}_2)$ such that $\overline{\mathcal{Q}}_1$ is not contained in $\overline{\mathcal{Q}}_2$ relative to $\overline{\Sigma} \cup \{\overline{\alpha}_{Q(0)}\}$.*

Proof ($\Rightarrow$) Without loss of generality we may assume that we have a model $\mathcal{I}$ such that $\mathcal{I}(0) \models \Sigma$ and $\mathsf{ans}(\mathcal{Q}_1, \mathcal{I}(0)) \not\subseteq \mathsf{eval}(\mathcal{Q}_2, \mathcal{I}(0))$. Construct a quasimodel Q for $cl(\Sigma, \mathcal{Q}_1, \mathcal{Q}_2)$ as in the proof of Theorem C.1. To show that $\overline{\mathcal{Q}}_1$ is not contained in $\overline{\mathcal{Q}}_2$ relative to $\overline{\Sigma} \cup \{\overline{\alpha}_{Q(0)}\}$, it is enough to extend the (non-temporal) model $\mathcal{I}(0)$ to the new 'surrogate' atoms of the form $A_{C_1 \mathcal{U} C_2}$ and $A_{\chi_1 \mathcal{U} \chi_2}$ in accordance with their behaviour in $\mathcal{I}$ at time point 0: $A_{C_1 \mathcal{U} C_2}^{\mathcal{I}(0)} = (C_1 \mathcal{U} C_2)^{\mathcal{I}(0)}$ and

$$A_{\chi_1 \mathcal{U} \chi_2}^{\mathcal{I}(0)} = \begin{cases} \top & \text{if } \mathcal{I}(0) \models \chi_1 \,\mathcal{U}\, \chi_2 \\ \bot & \text{otherwise.} \end{cases}$$

($\Leftarrow$) is also proved similarly to Theorem C.1. The only difference is that now we select $\mathcal{J}(0)$ so that $\mathcal{J}(0) \models \overline{\Sigma} \wedge \overline{\alpha}_{Q(0)}$ and $\mathsf{eval}(\overline{\mathcal{Q}}_1, \mathcal{J}(0)) \not\subseteq \mathsf{ans}(\overline{\mathcal{Q}}_2, \mathcal{J}(0))$. ❑

So, the query-containment problem for $\mathcal{DLR}_{\mathcal{U}}^{-}$ reduces to satisfiability in quasimodels and the query-containment problem for (non-temporal) $\mathcal{DLR}$. The latter problem was shown to be decidable in 2EXPTIME time in [Calvanese *et al.*, 1998a]. But how to check satisfiability in quasimodels? First of all, we need a procedure deciding whether a quasistate candidate is a quasistate for a given set of formulas and concepts. The following proposition can be proved using the reduction in[Calvanese *et al.*, 1998a].

Proposition C.4. (i) *Given a $\mathcal{DLR}_{\mathcal{U}}^{-}$-formula φ, it is decidable in NEXPTIME whether a quasistate candidate for $cl(\varphi)$ is a quasistate.*

(ii) *Given a $\mathcal{DLR}_{\mathcal{U}}^{-}$-schema Σ and queries $\mathcal{Q}_1$, $\mathcal{Q}_2$, it is decidable in 2EXPTIME whether $\overline{\mathcal{Q}}_1$ is contained in $\overline{\mathcal{Q}}_2$ relative to $\overline{\Sigma} \cup \{\overline{\alpha}_{\mathfrak{C}}\}$ for a quasistate candidate $\mathfrak{C}$ for $cl(\Sigma, \mathcal{Q}_1, \mathcal{Q}_2)$.*

Now, given a set Γ as defined above, we have at most $O(2^{2^{|\Gamma|}})$ distinct quasistates for Γ. The problem then is whether they can be properly arranged to form a quasimodel for Γ. As we have no past temporal operators, it is enough to consider the flow of time $\langle \mathbb{N}, < \rangle$ and quasimodels of the form $Q = \langle Q(n) : n \in \mathbb{N} \rangle$.

Let Q be a sequence of quasistates $Q(i) = \langle T_i, \Phi_i \rangle$, $i \in \mathbb{N}$, and r a sequence of elements from T_i such that $r(i) \in T_i$. Say that r *realises* $C\mathcal{U}D \in r(n)$ *in* m *steps* if there is $l \leq m$ such that $D \in r(n+l)$ and $C \in r(n+k)$ for all $k \in (0, l)$. A formula $\psi \mathcal{U} \chi \in \Phi_n$ is *realised in* m *steps* if there is $l \leq m$ such that $\chi \in \Phi_{n+l}$ and $\psi \in \Phi_{n+k}$ for all $k \in (0, l)$. We also say that a pair t, t' of concept types is *suitable* if for every $C\mathcal{U}D \in \Gamma$, $C\mathcal{U}D \in t$ iff either $D \in t'$ or $C \in t'$ and $C\mathcal{U}D \in t'$.

Suppose Q_1 and Q_2 are finite sequences of quasistates for Γ of length l_1 and l_2, respectively, and let $Q = Q_1 * Q_2^*$ (i.e., $Q = Q_1 * Q_2 * Q_2 * Q_2 * \dots$) with $Q(n) = \langle T_n, \Phi_n \rangle$. One can check that Q is a quasimodel for Γ if the following conditions hold:

(a) for every $i \leq l_1 + l_2$ and every $t' \in T_{i+1}$, there is $t \in T_i$ such that the pair t, t' is suitable;

(b) for every $i \leq l_1 + 1$ and every $t_i \in T_i$, all concepts of the form $C\mathcal{U}D \in t_i$ are realised in $l_1 + l_2 - i$ steps in some sequence $t_i, t_{i+1}, \dots, t_{l_1+l_2}$ in which $t_{i+j} \in T_{i+j}$ and every pair of adjacent elements is suitable;

(c) for every $i \leq l_1 + l_2$, and every $\psi \mathcal{U} \chi \in \Gamma$, $\psi \mathcal{U} \chi \in \Phi_i$ iff either $\chi \in \Phi_{i+1}$ or $\psi \in \Phi_{i+1}$ and $\psi \mathcal{U} \chi \in \Phi_{i+1}$;

(d) for every $i \leq l_1 + 1$, all formulas of the form $\psi \mathcal{U} \chi \in \Phi_i$ are realised in $l_1 + l_2 - i$ steps.

Moreover, given a quasimodel for Γ, one can always extract from it a subquasimodel $Q = Q_1 * Q_2^*$ which satisfies (a)–(d) above, all quasistates in Q_2 are distinct and $|Q_2| = O(2^{2^{|\Gamma|}})$.

Using this observation together with Proposition C.4 one can construct an EXPSPACE formula-satisfiability checking algorithm and a 2EXPTIME query-containment checking algorithm. A proof of EXPSPACE-hardness of the formula-satisfiability problem (for a much weaker logic) can be found at `http://www.dcs.kcl.ac.uk/staff/mz`. It follows, in particular, that the query-containment problem is EXPSPACE-hard as well. It is an open problem, however, whether there exists an EXPSPACE algorithm deciding this problem.

Finally, we show EXPTIME-completeness of the logical implication for atomic formulas in $\mathcal{DLR}_{\mathcal{U}}^{-}$ by means of a polynomial reduction of $\mathcal{DLR}_{\mathcal{U}}^{-}$ to the logic $\mathcal{DLR}_{reg}$ of [Calvanese *et al.*, 1998a]. For our purposes, it is enough to know that $\mathcal{DLR}_{reg}$ allows one to form the transitive closure R^* of every binary relation R, and that the satisfiability problem in $\mathcal{DLR}_{reg}$ is in EXPTIME. To simplify presentation, we reduce here the fragment $\mathcal{DLR}_{\Box\bigcirc}^{-}$ of $\mathcal{DLR}_{\mathcal{U}}^{-}$ with the temporal operators $\Box$ and $\bigcirc$ only (the reader should not have problems to extend this reduction to the language with $\mathcal{U}$).

Fix a binary relation R and define a translation $\star$ from $\mathcal{DLR}_{\Box\bigcirc}^{-}$ to $\mathcal{DLR}_{reg}$ as follows: $P^\star = P$ for every atom P of $\mathcal{DLR}$,

$$\begin{aligned} (\bigcirc C)^\star &= \forall R.C^\star \\ (\Box C)^\star &= \forall R^*.C^\star; \end{aligned}$$

$\star$ commutes with the remaining constructs, and $(P_1 \sqsubseteq P_2)^\star = P_1^\star \sqsubseteq P_2^\star$.

Lemma C.5. *Suppose that $\Gamma \cup \{\varphi\}$ is a set of atomic $\mathcal{DLR}^-_{\bigcirc\Box}$-formulas and that R does not occur in $\Gamma \cup \{\varphi\}$. Then $\Gamma \models \varphi$ iff $\varphi^\star$ is a logical consequence of the following set Ξ of $\mathcal{DLR}_{reg}$-formulas*

$$\Gamma^\star,\ \exists^{=1}R.\top \doteq \top,\ \exists^{=1}R^-.\top \doteq \top,$$

where $\exists^{=1}R^-.C = \exists^{=1}[2](R \sqcap 1/2 : C)$.

Proof Suppose $\Gamma \not\models \varphi$. Then there is a model $\mathcal{I}$ such that $\mathcal{I}, 0 \not\models \varphi$, but $\mathcal{I}, n \models \Gamma$ for all $n \in \mathbb{Z}$. Define a $\mathcal{DLR}$-model

$$\mathcal{J} = \left\langle \Delta', P_1^{\mathcal{J}}, \ldots, R^{\mathcal{J}} \right\rangle$$

by taking

- $\Delta' = \Delta^{\mathcal{I}} \times \mathbb{Z}$,
- $\langle \langle x_1, n_1 \rangle, \ldots, \langle x_l, n_l \rangle \rangle \in P_i^{\mathcal{J}}$ iff $n_i = n_j$, for $i, j \leq l$, and $\langle x_1, \ldots, x_n \rangle \in P_i^{\mathcal{I}(n_1)}$,
- $\langle \langle x_1, n_1 \rangle, \langle x_2, n_2 \rangle \rangle \in R^{\mathcal{J}}$ iff $x_1 = x_2$ and $n_2 = n_1 + 1$.

It is readily checked $\mathcal{J} \models \Xi$ and $\mathcal{J} \not\models \varphi^\star$.

Conversely, suppose that $\mathcal{J} = \left\langle \Delta, P_1^{\mathcal{J}}, \ldots, R^{\mathcal{J}} \right\rangle$ is a model such that $\mathcal{J} \models \Xi$ but $\mathcal{J} \not\models \varphi^\star$. Let $\Sigma = \bigcup\{cl(\chi) : \chi \in \Gamma \cup \{\varphi\}\}$ and, for every $x \in \Delta$,

$$t(x) = \{C \in c(\Sigma) : x \in (C^\star)^{\mathcal{J}}\}.$$

Then the pair $\langle T, \Phi \rangle$, where $T = \{t(x) : x \in \Delta\}$ and $\Phi = \{\chi \in f(\Sigma) : \mathcal{J} \models \chi^\star\}$, is a quasistate for Σ. Define a map Q by taking $Q(n) = \langle T, \Phi \rangle$ for all $n \in \mathbb{Z}$. It is easy to see that Q is a quasimodel. Hence, by Theorem C.1, we have a model $\mathcal{I}$ such that $\mathcal{I} \models \Gamma$ but $\mathcal{I}, 0 \not\models \varphi$. ❑