

Reasoning on temporal class diagrams: Undecidability results — [Source link](#)

Alessandro Artale

Institutions: Free University of Bozen-Bolzano

Published on: 01 Mar 2006 - Annals of Mathematics and Artificial Intelligence (Springer Netherlands)

Topics: Interval temporal logic, Class diagram, Knowledge representation and reasoning, Communication diagram and Description logic

Related papers:

- [A temporal description logic for reasoning about actions and plans](#)
- [The Description Logic Handbook: Theory, Implementation and Applications](#)
- [Executable temporal logic for non-monotonic reasoning](#)
- [Introducing temporal description logics](#)
- [Dynamic Linear Time Temporal Description Logics](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/reasoning-on-temporal-class-diagrams-undecidability-results-r0185a8inh>

Reasoning on Temporal Class Diagrams: Undecidability Results

Alessandro Artale

Dept. of Computer Science – Free Univeristy of Bozen-Bolzano.

E-mail: artale@inf.unibz.it

This paper introduces a temporal class diagram language useful to model temporal varying data. The atemporal portion of the language contains the core constructors available in both EER diagrams and UML class diagrams. The temporal part of the language is able to distinguish between temporal and atemporal constructs, and it has the ability to represent dynamic constraints between classes. The language is characterized by a model-theoretic (temporal) semantics. Reasoning services as logical implication and satisfiability are also defined. We show that reasoning on finite models is different from reasoning on unrestricted ones. Then, we prove that reasoning on temporal class diagrams is an undecidable problem on both unrestricted models and on finite ones.

Keywords: Temporal Data Models, Description Logics, Temporal Logics.

AMS Subject classification: ???

1. Introduction

Conceptual data modeling describes an application domain in a declarative and reusable way by specifying constraints on the use of the data and possibly drawing new information from it. Recently, a number of conceptual modeling languages has emerged as de facto standards; in particular, we mention entity-relationship (ER) for the relational data model, UML and ODMG for the object-oriented data model, and RDF, DAML+OIL and OWL for the web ontology languages. In this paper we deal with temporally extended conceptual data models developed to abstract the temporal aspects of information. Many temporal models have been developed (in particular to help designing temporal databases) and a summary of results achieved in the area can be found in two good surveys [14,17].

Here we propose a temporal class diagram formalism equipped with both a linear and a graphical syntax along with a model-theoretic semantics. The atemporal portion of the language contains the core constructors available in most of the conceptual models mentioned above. Essentially, *Classes* and *Relationships* are first class citizens. Classes can be organized in disjoint and/or covering generalized hierarchies. Relations between classes are modeled through n-ary relationships. Full cardinality constraints can be specified on the participation of classes into relationships. The temporal part of the language supports valid time for classes and relationships in the line of TIMEER [13] and ERT [20], while supporting dynamic constraints for classes as presented in MADS [19].

This paper moves from previous works of the author where the temporal conceptual model \mathcal{ER}_{VT} has been formally characterized [1,4]. Starting from such formalization we devise here a temporal modeling language as a sub-language of \mathcal{ER}_{VT} with the main intention to investigate whether reasoning over temporal diagrams is decidable. In addition to the classical EER constructors (the Extended Entity-Relationship data model, see [11]), the language proposed here is able to express the following temporal constraints:

- *Timestamping.* The data model distinguishes between *snapshot* constructs—i.e. each

of their instances has a global lifespan—and *temporary* constructs—i.e. each of their instances have a limited lifespan.

- *Dynamic Constructs.* They apply to classes by capturing the *object migration* from a source class to a target class. They are also called *transition* constraints [15] and they describe how an object can change its class membership from one class to another. For example, an object in the Employee class may later migrate to become an object of the Manager class.

The main result illustrated here is that reasoning on temporal conceptual models is undecidable providing the diagrams are able both: (a) to distinguish between temporal and non-temporal constructs; and (b) to represent *dynamic constraints* between classes, i.e. classes whose instances migrate to other classes. This result is different from a similar one presented in [5]. Indeed, in [5] the authors showed that temporal diagrams expressed in the \mathcal{ER}_{VT} modeling language can be embedded into the temporal description logic \mathcal{DLR}_{US} —where \mathcal{U}, \mathcal{S} extend \mathcal{DLR} [8] with the *until* and *since* temporal modalities—and that reasoning in \mathcal{DLR}_{US} was undecidable. On the other hand, here we prove that even reasoning just on temporal class diagrams (and thus on \mathcal{ER}_{VT} schemas) is undecidable. The undecidability result is proved via a reduction of the Halting Problem. In particular, we proceed by first showing that the halting problem can be encoded as a Knowledge Base (KB) in \mathcal{ALC}_F —where F extends the description logic \mathcal{ALC} with the *future* temporal modality—and then proving that such a KB in \mathcal{ALC}_F can be captured by a schema expressed in our temporal class diagram. Note that, in [12] the undecidability of \mathcal{ALC}_F is proved using both: (a) complex axioms—i.e. axioms can be combined using boolean and modal operators; and (b) both *global* and *local* axioms—i.e. axioms can be either true at all time or true at some time, respectively. Since the temporal class diagram is able to encode just simple global axioms, we modify the proof presented in [12] by showing that checking concept satisfiability w.r.t. an \mathcal{ALC}_F KB made by just simple global axioms is an undecidable problem. This new result on temporal description logics reduces the gap between decidable and undecidable languages.

We also show that temporal class diagrams do not enjoy the *Finite Model Property* (FMP) (that is, a class could be satisfiable only in models with an infinite domain). The negative undecidability result also holds when reasoning is restricted to finite models. Still the halting problem can be reduced to satisfiability of temporal class diagrams in finite models.

The paper is organized as follow. The temporal description logic \mathcal{ALC}_F is briefly introduced in Section 2. Section 3 gives a formal presentation of temporal class diagrams along with a running example. The reasoning services over temporal class diagrams are defined in Section 4. That reasoning in presence of dynamic constraints is undecidable in both unrestricted and finite models is proved in Sections 5 and 6, respectively. Section 7 makes final conclusions and summarizes the complexity results already obtained when dealing with temporal data languages with different expressivity power. In the conclusions an interesting open problem related to the complexity of reasoning is finally mentioned.

2. The Temporal Description Logic

In this Section we introduce the \mathcal{ALC}_F description logic [2,12,21] as a the tense-logical extension of the description logic \mathcal{ALC} . With respect to the formal apparatus, we will strictly follow the concept language formalism presented in [6]. In this perspective, description logics are considered a *structured* fragment of predicate logic. Basic types of \mathcal{ALC}_F are *concepts* and *roles*. A concept is a description gathering the common properties among a collection of individuals; from a logical point of view it is a unary predicate ranging over the domain of individuals. Inter-relationships between these individuals are

$C, D \rightarrow A$	(atomic concept)	$A^{\mathcal{I}(t)} \subseteq \Delta^{\mathcal{I}}$
\top	(top)	$\top^{\mathcal{I}(t)} = \Delta^{\mathcal{I}}$
\perp	(bottom)	$\perp^{\mathcal{I}(t)} = \emptyset$
$\neg C$	(complement)	$(\neg C)^{\mathcal{I}(t)} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}(t)}$
$C \sqcap D$	(conjunction)	$(C \sqcap D)^{\mathcal{I}(t)} = C^{\mathcal{I}(t)} \cap D^{\mathcal{I}(t)}$
$C \sqcup D$	(disjunction)	$(C \sqcup D)^{\mathcal{I}(t)} = C^{\mathcal{I}(t)} \cup D^{\mathcal{I}(t)}$
$\exists R.C$	(exist. quantifier)	$(\exists R.C)^{\mathcal{I}(t)} = \{a \in \Delta^{\mathcal{I}} \mid \exists b. R^{\mathcal{I}(t)}(a, b) \Rightarrow C^{\mathcal{I}(t)}(b)\}$
$\forall R.C$	(univ. quantifier)	$(\forall R.C)^{\mathcal{I}(t)} = \{a \in \Delta^{\mathcal{I}} \mid \exists b. R^{\mathcal{I}(t)}(a, b) \wedge C^{\mathcal{I}(t)}(b)\}$
$\diamond^+ C$	(Sometime in the Future)	$(\diamond^+ C)^{\mathcal{I}(t)} = \{a \in \Delta^{\mathcal{I}} \mid \exists v > t. C^{\mathcal{I}(v)}(a)\}$
$\square^+ C$	(Every time in the Future)	$(\square^+ C)^{\mathcal{I}(t)} = \{a \in \Delta^{\mathcal{I}} \mid \forall v > t. C^{\mathcal{I}(v)}(a)\}$

Figure 1. Syntax and Semantics for the \mathcal{ALCF} Description Logic

represented by means of roles, which are interpreted as binary relations over the domain of individuals. According to the syntax rules of Figure 1, \mathcal{ALCF} *concepts* (denoted by the letters C and D) are built out of *atomic concepts* (denoted by the letter A) and *atomic roles* (denoted by the letter R). Tense operators are added for concepts: \diamond^+ (sometime in the future) and \square^+ (always in the future). Furthermore, while tense operators are allowed only at the level of concepts—i.e. no temporal operators are allowed on roles—we will distinguish between so called *local*— \mathcal{RL} —and *global*— \mathcal{RG} —roles.

Let us now consider the formal semantics of \mathcal{ALCF} . A temporal structure $\mathcal{T} = (\mathcal{T}_p, <)$ is assumed, where \mathcal{T}_p is a set of time points and $<$ is a strict linear order on \mathcal{T}_p — \mathcal{T} is assumed to be isomorphic to either $(\mathbb{Z}, <)$ or $(\mathbb{N}, <)$. An \mathcal{ALCF} *temporal interpretation* over \mathcal{T} is a triple 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 concept C , and every role R , we have $C^{\mathcal{I}(t)} \subseteq \Delta^{\mathcal{I}}$ and $R^{\mathcal{I}(t)} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Furthermore, if $R \in \mathcal{RG}$, then, $\forall t_1, t_2 \in \mathcal{T}. R^{\mathcal{I}(t_1)} = R^{\mathcal{I}(t_2)}$. The semantics of concepts is defined in the right side of Figure 1—note that the operator \square^+ is the dual of \diamond^+ , i.e. $\square^+ C \equiv \neg \diamond^+ \neg C$.

A *knowledge base* (KB) in this context is a finite set Σ of *terminological axioms* of the form $C \sqsubseteq D$. An interpretation \mathcal{I} satisfies $C \sqsubseteq D$ if and only if the interpretation of C is included in the interpretation of D at all time, i.e. $C^{\mathcal{I}(t)} \subseteq D^{\mathcal{I}(t)}$, for all $t \in \mathcal{T}$. Thus, axioms are (a) *global*, since they must hold at *all time*; and (b) *simple*, since we cannot compose them with either boolean or modal operators—we call them *simple global axioms*. A knowledge base Σ is *satisfiable* if there is a temporal interpretation \mathcal{I} which satisfies every axiom in Σ ; in this case \mathcal{I} is called a *model* of Σ . Σ *logically implies* an axiom $C \sqsubseteq D$ (written $\Sigma \models C \sqsubseteq D$) if $C \sqsubseteq D$ is satisfied by every model of Σ . In this latter case, the concept C is said to be *subsumed* by the concept D in the knowledge base Σ . A concept C is *satisfiable*, given a knowledge base Σ , if there exists a model \mathcal{I} of Σ such that $C^{\mathcal{I}(t)} \neq \emptyset$ for some $t \in \mathcal{T}$, i.e. $\Sigma \not\models C \sqsubseteq \perp$.

As an example of a concept using temporal operators, consider the definition of a “person”. The class of persons denotes individuals who are currently living beings, live in some place, and eventually they will stop being living beings forever:

$$\text{Person} \sqsubseteq \text{LivingBeing} \sqcap \forall \text{LIVES-IN.Place} \sqcap \\ \diamond^+(\neg \text{LivingBeing} \sqcap \square^+ \neg \text{LivingBeing})$$

3. Temporal Class Diagrams

In this Section, the language to describe temporal class diagrams is introduced. Since we are interested in characterizing the source of undecidability, the language we study is

a proper subset of \mathcal{ER}_{VT} [4,1]¹. The main constructors are *Classes*—denoting set of objects—and *Relationships*—linking two classes. Classes can be organized in taxonomic hierarchies. Hierarchies could be total or partial, and overlapping or disjoint as in the EER (Extended Entity-Relationships) model [11]. Cardinality constraints specify the participation of classes into relationships. As far as temporal constructs are concerned, the language supports *timestamping* for both classes and relationships in the line of TIMEER [13] and ERT [20], while supporting *dynamic constraints* for classes as presented in MADS [19]. Since this work considers just *validity time* rather than transaction time (see the consensus glossary [16] for the terminology used), then the timestamping constructs model the notion of *lifespan* in temporal databases, i.e. the points in time when an object or a tuple belongs to a class or a relationship, respectively. In particular, the language is able to distinguish between *snapshot* constructs—i.e. constructs which bear no explicit specification of a given lifespan which we convey by assuming a global lifespan associated to each of their instances—*temporary* constructs—i.e. each of their instances have a limited lifespan—or *mixed* constructs—i.e. each of their instances can have either a global or a temporary existence. Two temporal marks, S and T are introduced to capture snapshot and temporary constructs, respectively. On the other hand, mixed constructs are left unmarked meaning that the modeler does not want to temporally constraint the construct. As logical implication is formally defined in \mathcal{ER}_{VT} (see Definition 4.1), missing constraints referring to timestamping can be inferred (see the running example below).

Dynamic constructs [15,19,4] (also called *transition constraints*) have been introduced to model the phenomenon called *object migration*. A transition models objects migrating from a *source* class to a *target* class. At the schema level, it expresses that the instances of the source class may *migrate* into the target class. A *dynamic extension* between a source and a target class (represented by a dotted link labeled with DEX) models the case where instances of the source class *eventually* become instances of the target class. On the other hand, a *dynamic persistency* (represented by a dotted link labeled with PER) models the dual case of instances *persistently* migrating to a target class. Before showing an example of a temporal class diagram let us introduce the syntax and the semantics of the language.

The language is equipped with both a linear and a graphical syntax along with a model-theoretic semantics as a temporal extension of the EER semantics [9]. Presenting the linear syntax, we adopt the following notation: given two sets X, Y , an *X-labeled* tuple over Y is a function from X to Y ; the labeled tuple T that maps the set $\{x_1, \dots, x_n\} \subseteq X$ to the set $\{y_1, \dots, y_n\} \subseteq Y$ is denoted by $\langle x_1 : y_1, \dots, x_n : y_n \rangle$, and $T[x_i] = y_i$. In the following definition we refer to Figure 2 as an example of the visual syntax² associated to the various constructs.

Definition 3.1. (Syntax) A diagram (also called schema) is a tuple: $\Sigma = (\mathcal{L}, \text{REL}, \text{CARD}, \text{ISA}, \text{DISJ}, \text{COVER}, \text{DEX}, \text{PER})$, such that:

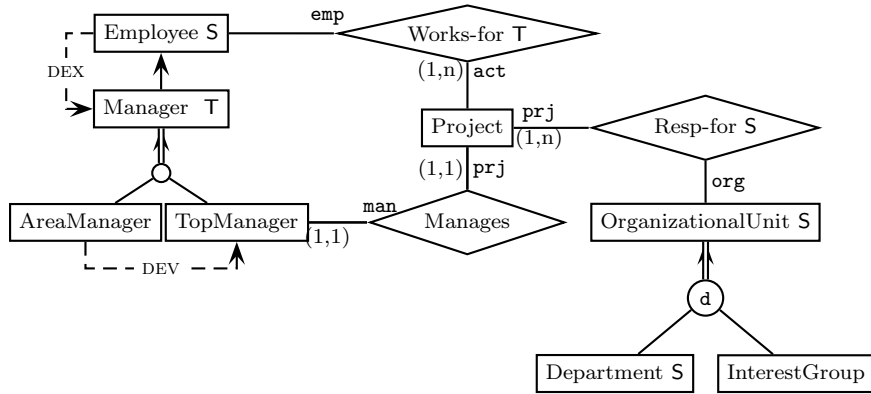
- \mathcal{L} is a finite alphabet partitioned into
 - * a set \mathcal{C} of *class* symbols,
 - * a set \mathcal{R} of *relationship* symbols,
 - * a set \mathcal{U} of *role* symbols.

We will call the tuple $(\mathcal{C}, \mathcal{R}, \mathcal{U})$ the *signature* of the schema Σ .

- \mathcal{C} is partitioned into

¹ As for the adequacy of \mathcal{ER}_{VT} for temporal conceptual modeling see [4].

² We adopt an EER style where classes are in boxes and relationships inside diamonds, ISA are directed lines, generalized hierarchies could be disjoint (circle with a 'd' inside) or covering (double directed lines).

Figure 2. The *Company* temporal diagram

- * a set \mathcal{C}^S of *snapshot classes* (the S-marked classes in the graphical representation of Fig. 2);
- * a set \mathcal{C}^M of *Mixed temporal classes* (the unmarked classes);
- * and a set \mathcal{C}^T of *temporary classes* (the T-marked classes).

- A similar partition applies to the set \mathcal{R} .
- REL is a function that maps a relationship symbol in \mathcal{R} to an \mathcal{U} -labeled tuple over \mathcal{C} ,

$$\text{REL}(R) = \langle U_1 : C_1, \dots, U_k : C_k \rangle$$

and k is the *arity* of R .

- CARD is a function

$$\mathcal{C} \times \mathcal{R} \times \mathcal{U} \mapsto \mathbb{N} \times (\mathbb{N} \cup \{\infty\})$$

denoting cardinality constraints.

If $\text{REL}(R) = \langle U_1 : C_1, \dots, U_k : C_k \rangle$, then, $\text{CARD}(C, R, U)$ is defined only if $U = U_i$ and $C = C_i$, for some $i \in \{1, \dots, k\}$. We denote with $\text{CMIN}(C, R, U)$ and $\text{CMAX}(C, R, U)$ the first and second component of CARD. If not stated otherwise, CMIN is assumed to be zero, and CMAX is assumed to be ∞ .

- ISA is a binary relationship

$$\text{ISA} \subseteq (\mathcal{C} \times \mathcal{C}) \cup (\mathcal{R} \times \mathcal{R}).$$

ISA between relationships is restricted to relationships with the same arity.³

- DISJ, COVER are binary relations over $2^{\mathcal{C}} \times \mathcal{C}$, describing disjointness and covering partitions between classes, respectively.
- Both DEX and PER are binary relations over $\mathcal{C} \times \mathcal{C}$ describing the evolution of classes.

The model-theoretic semantics associated with the language adopts the *snapshot*⁴ representation of abstract temporal databases and temporal conceptual models [10]. Following this paradigm, the flow of time $\mathcal{T} = \langle \mathcal{T}_p, < \rangle$, where \mathcal{T}_p is a set of time points (or chronons) and $<$ is a binary precedence relation on \mathcal{T}_p , is assumed to be isomorphic to ei-

³ For ISA relations, we use the notation $C_1 \text{ ISA } C_2$ instead of $\langle C_1, C_2 \rangle \in \text{ISA}$. Similarly for DISJ, COVER, DEX, PER.

⁴ The snapshot model represents the same class of temporal databases as the *timestamp* model [17,18] defined by adding temporal attributes to a relation [10].

ther $\langle \mathbb{Z}, < \rangle$ or $\langle \mathbb{N}, < \rangle$. 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 domain.

Definition 3.2. (Semantics) Let Σ be a schema. A *temporal database state* for the schema Σ is a tuple $\mathcal{B} = (\mathcal{T}, \Delta^{\mathcal{B}}, \cdot^{\mathcal{B}(t)})$, such that: $\Delta^{\mathcal{B}}$ is a nonempty set; $\cdot^{\mathcal{B}(t)}$ is a function such that for each $t \in \mathcal{T}$, every class $C \in \mathcal{C}$, and every relationship $R \in \mathcal{R}$, we have: $C^{\mathcal{B}(t)} \subseteq \Delta^{\mathcal{B}}$, while $R^{\mathcal{B}(t)}$ is a set of \mathcal{U} -labeled tuples over $\Delta^{\mathcal{B}}$.

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

- For each $C_1, C_2 \in \mathcal{C}$, if C_1 ISA C_2 , then, $C_1^{\mathcal{B}(t)} \subseteq C_2^{\mathcal{B}(t)}$.
- For each $R_1, R_2 \in \mathcal{R}$, if R_1 ISA R_2 , then, $R_1^{\mathcal{B}(t)} \subseteq R_2^{\mathcal{B}(t)}$.
- For each cardinality constraint $\text{CARD}(C, R, U)$, then, $e \in C^{\mathcal{B}(t)} \rightarrow \text{CMIN}(C, R, U) \leq \#\{r \in R^{\mathcal{B}(t)} \mid r[U] = e\} \leq \text{CMAX}(C, R, U)$.
- For each snapshot class $C \in \mathcal{C}^S$, then, $e \in C^{\mathcal{B}(t)} \rightarrow \forall t' \in \mathcal{T}. e \in C^{\mathcal{B}(t')}$.
- For each temporary class $C \in \mathcal{C}^T$, then, $e \in C^{\mathcal{B}(t)} \rightarrow \exists t' \neq t. e \notin C^{\mathcal{B}(t')}$.
- For each snapshot relationship $R \in \mathcal{R}^S$, then, $r \in R^{\mathcal{B}(t)} \rightarrow \forall t' \in \mathcal{T}. r \in R^{\mathcal{B}(t')}$.
- For each temporary relationship $R \in \mathcal{R}^T$, then, $r \in R^{\mathcal{B}(t)} \rightarrow \exists t' \neq t. r \notin R^{\mathcal{B}(t')}$.
- For $C, C_1, \dots, C_n \in \mathcal{C}$,
 - * If $\{C_1, \dots, C_n\}$ DISJ C , then, $\forall i \in \{1, \dots, n\}. C_i$ ISA $C \wedge \forall j \in \{1, \dots, n\}, j \neq i. C_i^{\mathcal{B}(t)} \cap C_j^{\mathcal{B}(t)} = \emptyset$.
 - * If $\{C_1, \dots, C_n\}$ COVER C , then, $\forall i \in \{1, \dots, n\}. C_i$ ISA $C \wedge C^{\mathcal{B}(t)} = \bigcup_{i=1}^n C_i^{\mathcal{B}(t)}$.
- For each $C_1, C_2 \in \mathcal{C}$,
 - * If C_1 DEX C_2 , then, $e \in C_1^{\mathcal{B}(t)} \rightarrow \exists t_1 > t. e \in C_2^{\mathcal{B}(t_1)}$;
 - * If C_1 PER C_2 , then, $e \in C_1^{\mathcal{B}(t)} \rightarrow \forall t' > t. e \in C_2^{\mathcal{B}(t')}$.

In addition to dynamic extensions, another form of migration has been studied in the literature. *Dynamic evolution* is a transition where the migrating object ceases to be an instance of the source class. We also consider here a *strong dynamic evolution* where the migrating object will never go back to the source class. We give a formal definition of these constraints and then show that DEX and PER are sufficient to capture them.

Definition 3.3. (Evolution) We consider two forms of evolution between classes: *dynamic evolution* (DEV) and *strong dynamic evolution* (S-DEV) with the following semantics:

- **Dynamic Evolution.**
If C_1 DEV C_2 , then, $e \in C_1^{\mathcal{B}(t)} \rightarrow \exists t_1 > t. (e \in C_2^{\mathcal{B}(t_1)} \wedge e \notin C_1^{\mathcal{B}(t_1)})$
- **Strong Dynamic Evolution.**
If C_1 S-DEV C_2 , then, $e \in C_1^{\mathcal{B}(t)} \rightarrow \exists t_1 > t. (e \in C_2^{\mathcal{B}(t_1)} \wedge \forall t_2 \geq t_1. e \notin C_1^{\mathcal{B}(t_2)})$

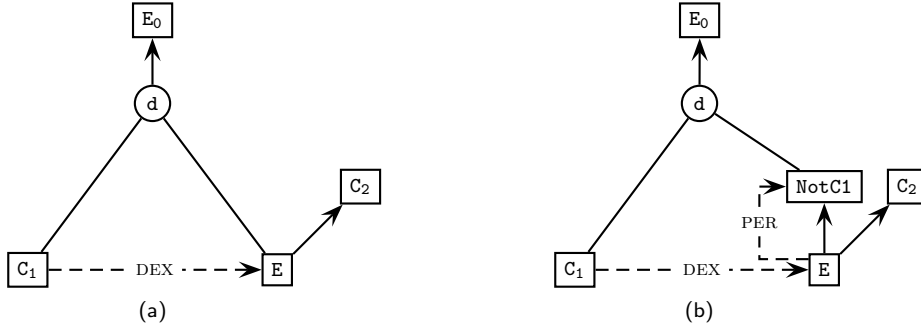


Figure 3. Encoding (a) dynamic and (b) strong dynamic evolution.

The following Proposition shows that both DEV and S-DEV can be modeled by using both the persistency and dynamic extension constructors—i.e. they do not add further expressivity to our language as defined in Definition 3.1.

Proposition 3.4. The diagram of Figure 3(a) is a correct mapping of C_1 DEV C_2 . By adding to the diagram the constraint E PER NotC1 (Figure 3(b)) we encode the S-DEV constructor.

Proof. (DEV) Let $e \in C_1^{\mathcal{B}(t)}$, then, by (DEX) $\exists t_1 > t. e \in E^{\mathcal{B}(t_1)}$. Thus, by (ISA) $e \in C_2^{\mathcal{B}(t_1)}$ and since E is disjoint from C_1 , $e \notin C_1^{\mathcal{B}(t_1)}$.

(S-DEV) As for DEV, $e \in C_2^{\mathcal{B}(t_1)}$ and $e \notin C_1^{\mathcal{B}(t_1)}$. Furthermore, by (PER), $\forall t_2 > t_1. e \in \text{NotC1}^{\mathcal{B}(t_2)}$, i.e. $\forall t_2 > t_1. e \notin C_1^{\mathcal{B}(t_2)}$. \square

3.1. A Running Example

The various components of a temporal class diagram are now illustrated with respect to the *Company* schema of Figure 2. We start by showing the alphabet of the example schema. The sets of snapshot classes and relationships are:

$$\begin{aligned} \mathcal{C}^S &= \{\text{Employee}, \text{Department}, \text{OrganizationalUnit}\}, \\ \mathcal{R}^S &= \{\text{Resp-for}\}. \end{aligned}$$

The sets of temporary classes and relationships are:

$$\begin{aligned} \mathcal{C}^T &= \{\text{Manager}\}, \\ \mathcal{R}^T &= \{\text{Works-for}\}. \end{aligned}$$

The set of mixed classes and relationships are:

$$\begin{aligned} \mathcal{C}^M &= \{\text{AreaManager}, \text{TopManager}, \text{InterestGroup}, \text{Project}\} \\ \mathcal{R}^M &= \{\text{Manages}\}. \end{aligned}$$

Instances of temporary classes and relationships are intended to have a limited lifespan. On the other hand, the set of instances of snapshot classes and relationships never changes in time. For mixed classes and relationships, no temporal constraint holds—i.e. their set of instances can contain either instances with a limited lifespan or instances with unlimited lifespan. Note that, to capture the semantics of a *legacy* diagram, each of its elements should be necessarily snapshot marked. This preprocessing step is necessary to enforce *upward compatibility* when *legacy* diagrams are included in a temporal diagram.

The function REL associates with each argument of a relationship a name, called *role*, and a class describing its type, for example

$\text{REL}(\text{Manages}) = \langle \text{man} : \text{TopManager}, \text{prj} : \text{Project} \rangle$

describes **Manages** as a binary relationship, where a **TopManager** manages a **Project**. To associate cardinality constraints the function **CARD** can be used. For example:

$\text{CARD}(\text{TopManager}, \text{Manages}, \text{man}) = \langle 1, 1 \rangle$

states that a **TopManager** is constrained in managing exactly one project at a time.

ISA, **COVER**, and **DISJ** are used to represent generalized hierarchies. **ISA** models the subclass relationship, for example, **ManagerISAEmployee** says that **manager** is a subclass of **employee**. **COVER** models the fact that a set of subclasses may have common instances, but each instance of the superclass belongs to at least one of those subclasses as, for example, in:

$\{\text{AreaManager}, \text{TopManager}\} \text{COVER } \text{Manager}.$

DISJ models disjoint hierarchies, and:

$\{\text{Department}, \text{InterestGroup}\} \text{DISJ } \text{OrganizationalUnit}$

says that **department** is disjoint from **interest group** and both are subclasses of **organizational units**. By using both **COVER** and **DISJ** we can model *partitions*, i.e. disjoint and covering hierarchies:

$\{\text{Department}, \text{InterestGroup}\} \text{DISJ } \text{OrganizationalUnit}$
 $\{\text{Department}, \text{InterestGroup}\} \text{COVER } \text{OrganizationalUnit}.$

DEX, **DEV**, **S-DEV** and **PER** express dynamic constraints between classes. They could be used to model various form of *object migration* from a class to another. We can model the fact that a mere employee becomes a manager by defining a dynamic extension from the class **Employee** to the class **Manager**:

Employee DEX Manger

The following constraint

AreaManager DEV TopManger

says that an **AreaManager** will eventually become a **TopManager** while, since the two concepts are disjoint, the migration is actually an evolution. If we want to stress that an **AreaManger** will become a **TopManager** in the future but he will never be an **AreaManager** anymore then we will specify a strong evolution:

AreaManager S-DEV TopManger

The **PER** constructor can be used to express various forms of temporal constraints. The fact that a manager will always be a manager can be expressed as:

Manager PER Manger

Please, note that this constraint is consistent with the temporary marking associated to the manager class. Another use of **PER** is to avoid undesired transitions. For example, adding to the **Company** diagram the two disjoint classes **HumanBeing** and **MaterialObject** (an its complement **NotMaterialObject**) together with the hierarchical constraints: **Employee ISA HumanBeing**, and **OrganizationalUnit ISA MaterialObject** then we can forbid transitions from humans to material with the constraint:

HumanBeing PER NotMaterialObject

which would avoid altogether any transition from employees and its sub-classes to any of the organizational units of the company.

4. Reasoning on Temporal Models

Reasoning tasks over temporal class diagrams include verifying whether a class, relationship, or schema are *satisfiable*, whether a *subsumption* relation exists between classes or relationships, or checking whether a new schema property is *logically implied* by a given schema. The model-theoretic semantics associated with the class language allows us to formally define the reasoning tasks.

Definition 4.1. (Reasoning problems) Let Σ be a schema, $C \in \mathcal{C}$ a class, and $R \in \mathcal{R}$ a relationship. The following reasoning tasks over Σ can be defined:

1. $C (R)$ is *satisfiable* if there exists a legal temporal database state \mathcal{B} for Σ such that $C^{\mathcal{B}(t)} \neq \emptyset$ ($R^{\mathcal{B}(t)} \neq \emptyset$), for some $t \in \mathcal{T}$;
2. Σ is *satisfiable* if there exists a legal temporal database state \mathcal{B} for Σ that satisfies at least one class in Σ (\mathcal{B} is said a *model* for Σ);
3. $C_1 (R_1)$ is *subsumed* by $C_2 (R_2)$ in Σ if every legal temporal database state for Σ is also a legal temporal database state for $C_1 \text{ ISA } C_2$ ($R_1 \text{ ISA } R_2$);
4. A schema Σ' is *logically implied* by a schema Σ over the same signature if every legal temporal database state for Σ is also a legal temporal database state for Σ' .

Remark 4.2 Logical implication is the most powerful reasoning service. Indeed, checking whether a class C is satisfiable can be reduced to logical implication. By choosing $\Sigma' = \{C \text{ ISA } A, C \text{ ISA } B, \{A, B\} \text{ DISJ } O\}$, with A, B, O arbitrary classes, then C is satisfiable iff $\Sigma \not\models \Sigma'$. Furthermore, given two classes (relationships) C_1, C_2 (R_1, R_2), checking for subclass (subrelationship) can be reduced to logical implication by choosing $\Sigma' = \{C_1 \text{ ISA } C_2\}$ ($\Sigma' = \{R_1 \text{ ISA } R_2\}$).

Classical implications found in the literature of temporal conceptual modeling are captured by the logical implication service:

1. Subclasses of temporary classes are also temporary;
2. Subclasses of snapshot or mixed classes can be snapshot, temporary, or mixed classes;
3. If exactly one of a whole set of snapshot subclasses partitioning a snapshot superclass is temporary, then, the whole set of classes is unsatisfiable;
4. Participants of snapshot relations are either snapshot or mixed classes. They are snapshot when they participate at least once in the relationship;
5. Participants of temporary or mixed relations can be snapshot, temporary, or mixed classes;
6. A relationship is temporary if one of the participating classes is temporary.

Note that points 1 and 2 are also true for relationships.

Other reasoning problems could be defined in a temporal setting. For example, liveness- (i.e., infinitely often) and global-satisfiability (i.e., at all times) have been introduced in [4,3]. In this paper we concentrate on the core reasoning services and we then prove that even in such a setting complete automated reasoning is infeasible.

5. Reasoning on Temporal Class Diagrams is Undecidable

We now show that reasoning on temporal class diagrams is undecidable. The proof is based on a reduction from the undecidable halting problem for a Turing machine to the class satisfiability problem w.r.t. a schema Σ . We apply ideas similar to [12] (Sect. 7.5) to show undecidability of certain products of modal logics. The proof can be divided in the following steps:

1. Definition of the halting problem;
2. Reduction of the halting problem to concept satisfiability problem w.r.t. an \mathcal{ALCC}_F KB;
3. Reduction of concept satisfiability w.r.t. an \mathcal{ALCC}_F KB to class satisfiability w.r.t. a temporal class diagram.

The second step has been chosen as an intermediate step to better understand the halting problem reduction by using the concise \mathcal{ALCC}_F linear syntax. Then, the final step will show how temporal class diagrams are able to capture all the \mathcal{ALCC}_F axioms present in the reduction.

Halting problem

We show here a formal representation of the halting problem for Turing machines as presented in [12]. A single-tape right-infinite deterministic Turing machine \mathbf{M} is a triple $\langle A, S, \rho \rangle$, where: A is the *tape alphabet* ($b \in A$ stands for blank); S is a finite set of *states* with the *initial state*, s_0 , and the *final state*, s_1 ; ρ is the *transition function*, $\rho : (S - \{s_1\}) \times A \rightarrow S \times (A \cup \{\mathbf{L}, \mathbf{R}\})$. A *Configuration* of \mathbf{M} is an infinite sequence: $\langle \mathcal{L}, a_1, \dots, a_{i-1}, \langle s_i, a_i \rangle, \dots, a_n, b, \dots \rangle$, where, $\mathcal{L} \notin A$ is a symbol marking the left end of the tape, $a_i \in A$, and $s_i \in S$ is the current state. The cell $\langle s_i, a_i \rangle$ is the *active cell*. All the cells to the right of a_n are blank.

Since a transition function can only modify the active cell and its neighbors we introduce the *instruction function*, δ , defined on triples in $(A \cup \{\mathcal{L}\}) \times ((S - \{s_1\}) \times A) \times A$, such that:

$$\delta(a_i, \langle s, a_j \rangle, a_k) = \begin{cases} \langle a_i, \langle s', a'_j \rangle, a_k \rangle & \text{if } \rho(s, a_j) = \langle s', a'_j \rangle \\ \langle \langle s', a_i \rangle, a_j, a_k \rangle & \text{if } \rho(s, a_j) = \langle s', \mathbf{L} \rangle, \text{ and } a_i \neq \mathcal{L} \\ \langle \mathcal{L}, \langle s', a_j \rangle, a_k \rangle & \text{if } \rho(s, a_j) = \langle s', \mathbf{L} \rangle, \text{ and } a_i = \mathcal{L} \\ \langle a_i, a_j, \langle s', a_k \rangle \rangle & \text{if } \rho(s, a_j) = \langle s', \mathbf{R} \rangle \end{cases}$$

A sequence $\langle c_0, c_1, \dots, c_k, c_{k+1}, \dots \rangle$ of configurations of \mathbf{M} is said to be a *computation* of \mathbf{M} if the state of c_0 is s_0 (the initial state), and, for all k , c_{k+1} is obtained from c_k by replacing the triple centered around the active cell of c_k by its δ -image and leaving the rest unaltered. We say that \mathbf{M} *halts*, starting with the empty tape—i.e. with starting configuration: $\langle \mathcal{L}, \langle s_0, b \rangle, b, \dots, b, \dots \rangle$ —if there is a finite computation, $\langle c_0, c_1, \dots, c_k \rangle$, such that the state of c_k is s_1 (the final state).

Reasoning on \mathcal{ALCC}_F is undecidable

Using a reduction from the halting problem we now prove that reasoning involving an \mathcal{ALCC}_F knowledge base is undecidable. In [12] the undecidability of \mathcal{ALCC}_F is proved using: (a) complex axioms—i.e. axioms can be combined using Boolean and modal operators— (b) both *global* and *local* axioms—i.e. axioms can be either true at all time or true at some time, respectively. Since class diagrams are able to encode just simple global axioms, we modify the proof presented in [12]. The following theorem proves a new result for temporal description logics, i.e. that checking concept satisfiability w.r.t. an \mathcal{ALCC}_F KB made by just *simple global axioms* is an undecidable problem.

Theorem 5.1. Concept satisfiability w.r.t. an \mathcal{ALCC}_F knowledge base using just simple global axioms is undecidable.

Proof. Given a Turing machine, $\mathbf{M} = \langle A, S, \rho \rangle$, we construct an \mathcal{ALCC}_F KB, say \mathbf{KB}_M , with a concept that is satisfiable w.r.t. \mathbf{KB}_M iff the machine \mathbf{M} does not halt. We start by introducing some shortcuts. The implication, $C \rightarrow D$, is equivalent to the concept

expression $\neg C \sqcup D$. Given two concepts C, D we define $\mathbf{next}(C, D)$ as the following axiom: $C \sqsubseteq \diamond^+ D \sqcap \neg \diamond^+ D$. This axiom says that whenever $o \in C^{\mathcal{I}(t_0)}$, then, $o \in D^{\mathcal{I}(t_0+1)} \wedge \forall t \neq t_0. o \notin C^{\mathcal{I}(t)}$. Let C, D_1, \dots, D_n concepts, $\mathbf{discover}(C, \{D_1, \dots, D_n\})$ is defined as the conjunction of the following axioms:

$$\begin{aligned} C &\sqsubseteq D_1 \sqcup \dots \sqcup D_n \\ D_1 &\sqsubseteq C \sqcap \neg D_2 \sqcap \dots \sqcap \neg D_n \\ &\dots \\ D_{n-1} &\sqsubseteq C \sqcap \neg D_n \\ D_n &\sqsubseteq C \end{aligned}$$

i.e., there is a disjoint covering between C and $D_1 \dots D_n$.

Let $A' = A \cup \{\mathcal{L}\} \cup (S \times A)$. With each $x \in A'$ we introduce a concept C_x . We also use concepts C_s, C_l, C_r to denote the active cell, its left and right cells, respectively. The concept $S1$ denotes the final state. The halting problem reduces to satisfiability of C_0 . Extra concepts C, D_1, D_2, D_3 , will be also used. R is a global role. \mathbf{KB}_M contains the following axioms:

$$C_0 \sqsubseteq C_{\mathcal{L}} \sqcap \diamond^+ C_{\langle s_0, b \rangle} \quad (1)$$

$$\mathbf{discover}(C, \{C_x \mid x \in A'\}) \quad (2)$$

$$\top \sqsubseteq \exists R. \top \quad (3)$$

$$\mathbf{next}(C_{\mathcal{L}}, D_1) \quad (4)$$

$$\mathbf{next}(D_1, D_2) \quad (5)$$

$$C_{\langle s_0, b \rangle} \sqsubseteq D_1 \quad (6)$$

$$C_{\langle s_0, b \rangle} \sqsubseteq \square^+ C_b \quad (7)$$

$$\mathbf{discover}(C_s, \{C_{\langle s, a \rangle} \mid \langle s, a \rangle \in S \times A\}) \quad (8)$$

$$\mathbf{next}(C_l, C_s) \quad (9)$$

$$\mathbf{next}(C_s, C_r) \quad (10)$$

$$\mathbf{next}(C_r, D_3) \quad (11)$$

$$C_{\mathcal{L}} \sqsubseteq C_l \sqcup \diamond^+ C_l \quad (12)$$

$$C_l \sqsubseteq C_{\alpha} \rightarrow \forall R. C_{\alpha'} \quad (13)$$

$$C_s \sqsubseteq C_{\beta} \rightarrow \forall R. C_{\beta'} \quad (14)$$

$$C_r \sqsubseteq C_{\gamma} \rightarrow \forall R. C_{\gamma'} \quad (15)$$

$$C_a \sqsubseteq (\neg C_l \sqcap \neg C_s \sqcap \neg C_r) \rightarrow \forall R. C_a, \quad \forall a \in A \cup \{\mathcal{L}\} \quad (16)$$

$$\mathbf{discover}(S1, \{C_{\langle s_1, a \rangle} \mid a \in A \cup \{\mathcal{L}\}\}) \quad (17)$$

$$C_s \sqsubseteq \neg S1 \quad (18)$$

with axioms (13–15) for each instruction $\delta(\alpha, \beta, \gamma) = \langle \alpha', \beta', \gamma' \rangle$. We now prove that C_0 is satisfiable w.r.t. \mathbf{KB}_M iff \mathbf{M} has an infinite computation starting from the empty tape.

“ \Rightarrow ” Let C_0 be satisfiable, then, $\exists \langle x_0, t_0 \rangle \in \Delta^{\mathcal{I}} \times \mathcal{T}. x_0 \in C_0^{\mathcal{I}(t_0)}$. Then, by axiom (1), $x_0 \in C_{\mathcal{L}}^{\mathcal{I}(t_0)}$, and $\exists \bar{t} > t_0. x_0 \in C_{\langle s_0, b \rangle}^{\mathcal{I}(\bar{t})}$. We now show that $\bar{t} = t_0 + 1$. Indeed, if $C_{\langle s_0, b \rangle}$ is true, then, by axiom (6), D_1 must also be true, i.e. $x_0 \in D_1^{\mathcal{I}(\bar{t})}$. On the other hand, by axiom (4), $C_{\mathcal{L}}$ is true at just one point in time and D_1 is true next time and only there (by axiom (5)), i.e. $x_0 \in D_1^{\mathcal{I}(t_0+1)}$. Thus, $\bar{t} = t_0 + 1$, $x_0 \in C_{\mathcal{L}}^{\mathcal{I}(t_0)}$, $x_0 \in C_{\langle s_0, b \rangle}^{\mathcal{I}(t_0+1)}$, and, by axiom (7), $\forall t > t_0 + 1. x_0 \in C_b^{\mathcal{I}(t)}$. Furthermore, by axiom (8), $x_0 \in C_s^{\mathcal{I}(t_0+1)}$, while, by axioms (9–12), $x_0 \in C_l^{\mathcal{I}(t_0)}$, $x_0 \in C_r^{\mathcal{I}(t_0+2)}$. By axiom (2), for all $t \in \mathcal{T}$ there is at most one $x \in A'$ such that $x_0 \in C_x^{\mathcal{I}(t)}$, then, the sequence $\langle \langle x_0, t_0 \rangle, \langle x_0, t_0 + 1 \rangle, \dots \rangle$ represents

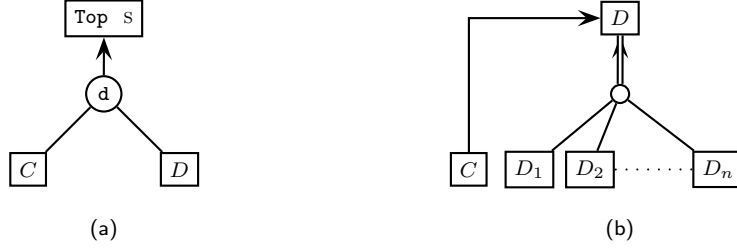


Figure 4. Encoding axioms: (a) $C \sqsubseteq \neg D$; (b) $C \sqsubseteq D_1 \sqcup \dots \sqcup D_n$.

the starting configuration of \mathbf{M} . Now, by axiom (3) and the assumption that R is global, $\exists x_1 \in \Delta^{\mathcal{I}}. \forall t \in \mathcal{T}. \langle x_0, x_1 \rangle \in R^{\mathcal{I}(t)}$ (we call x_1 R -successor of x_0). Let $\langle x_0, x_1, x_2, \dots \rangle$ be a chain of R -successors satisfying axiom (3). Since $x_0 \in C_{\mathcal{L}}^{\mathcal{I}(t_0)}$ and $x_0 \in C_l^{\mathcal{I}(t_0)}$, then, by axioms (13) and (16), and the definition of the instruction function, δ , $x_i \in C_{\mathcal{L}}^{\mathcal{I}(t_0)}$, for all i . Then, given the axioms (13–16), the chain of R -successor, $\langle x_0, x_1, x_2, \dots \rangle$, represents a computation of \mathbf{M} . Finally, axioms (17–18) guarantee that \mathbf{M} never halts.

“ \Leftarrow ” Conversely, suppose that \mathbf{M} is a Turing machine and $\langle c_0, \dots, c_k, \dots \rangle$ its infinite computation starting with the empty tape. We construct a model $\mathcal{I} \doteq \langle \mathcal{T}, \Delta^{\mathcal{I}}, \mathcal{I}^{\mathcal{I}(t)} \rangle$ of \mathbf{KB}_M such that C_0 is satisfiable. In particular, we fix $\mathcal{T} = \langle \mathbb{N}, < \rangle^5$, $\Delta^{\mathcal{I}} = \mathbb{N}$, $R^{\mathcal{I}} = \text{succ}_{\mathbb{N}}$ (the successor function over \mathbb{N}), $C_0^{\mathcal{I}(0)} = \{0\}$, and $C_0^{\mathcal{I}(j)} = \emptyset$, for all $j > 0$. Furthermore, $\forall j \in \mathbb{N}$:

- $C_x^{\mathcal{I}(j)} = \{i \in \mathbb{N} \mid \text{the } j\text{th cell of } c_i \text{ contains } x\}$, for all $x \in A'$
- $C_s^{\mathcal{I}(j)} = \{i \in \mathbb{N} \mid \text{the } j\text{th cell of } c_i \text{ is the active one}\}$
- $C_l^{\mathcal{I}(j)} = C_s^{\mathcal{I}(j+1)}$
- $C_r^{\mathcal{I}(j)} = C_s^{\mathcal{I}(j-1)}$
- $C^{\mathcal{I}(j)} = \bigcup_{x \in A'} C_x^{\mathcal{I}(j)}$
- $D_1^{\mathcal{I}(j)} = C_{\mathcal{L}}^{\mathcal{I}(j-1)}$
- $D_2^{\mathcal{I}(j)} = D_1^{\mathcal{I}(j-1)}$
- $D_3^{\mathcal{I}(j)} = C_r^{\mathcal{I}(j-1)}$
- $S1^{\mathcal{I}(j)} = \bigcup_{a \in A} C_{\langle s1, a \rangle}^{\mathcal{I}(j)}$.

It is easy to verify that \mathcal{I} is a model of \mathbf{KB}_M where C_0 is satisfiable. □

Reducing \mathcal{ALC}_F satisfiability to temporal class diagrams satisfiability

We now show how to capture the \mathcal{ALC}_F knowledge base \mathbf{KB}_M with a temporal class diagram, Σ_M . The mapping is based on a similar reduction presented in [7] for capturing \mathcal{ALC} axioms. For each atomic concept and role in \mathbf{KB}_M we introduce a class and a relationship, respectively. To simulate the universal concept, \top , we introduce a snapshot class, Top , that generalizes all the classes in Σ_M . Axioms of the form $C \sqsubseteq D_1 \sqcap D_2$ are replaced by two axioms $C \sqsubseteq D_1$, $C \sqsubseteq D_2$. Furthermore, axioms (12-16) have the general form $C \sqsubseteq C_1 \sqcup C_2$ with C_1, C_2 generic concept expressions. As proved in [7] they can be split by introducing new concept names $\overline{C}_1, \overline{C}_2$ as follows:

⁵ A similar proof holds if $\mathcal{T} = \langle \mathbb{Z}, < \rangle$.

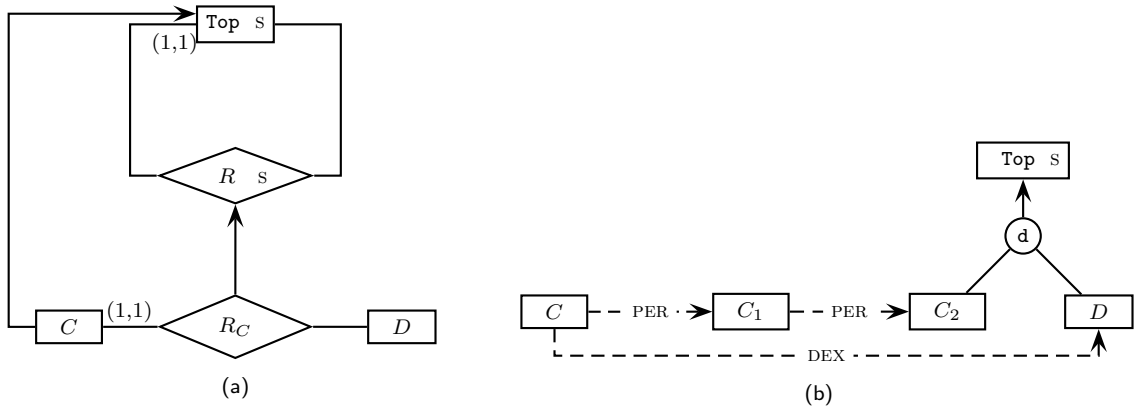


Figure 5. Encoding axioms: (a) $C \sqsubseteq \forall R.D$ and $\top \sqsubseteq \exists R.\top$; (b) $\text{next}(C, D)$.

$$\begin{aligned} C &\sqsubseteq \overline{C}_1 \sqcup \overline{C}_2 \\ \overline{C}_1 &\sqsubseteq C_1 \\ \overline{C}_2 &\sqsubseteq C_2 \end{aligned}$$

Given the various axioms in \mathbf{KB}_M , where the above equivalence-preserving translation has been applied, they are encoded as temporal diagrams as follows:

1. Axioms involving **discover** are mapped using disjoint and covering hierarchies.
2. Axioms of the form $C \sqsubseteq D$ (with C, D atomic concepts) are encoded as $C \text{ ISA } D$.
3. For each axiom of the form $C \sqsubseteq \neg D$ we construct the hierarchy in Figure 4(a).
4. For each axiom of the form $C \sqsubseteq D_1 \sqcup \dots \sqcup D_n$ we introduce a new class, D , and then we construct the hierarchy in Figure 4(b).
5. Axioms of the form $C \sqsubseteq \forall R.D$ are mapped together with the axiom $\top \sqsubseteq \exists R.\top$ by introducing a new sub-relationship, R_C , and considering R as a functional role⁶. Figure 5(a) shows the mapping where R is a snapshot relationship to capture the fact that R is a global role in \mathbf{KB}_M .
6. For each axiom of the form $C \sqsubseteq \square^+ D$ ($C \sqsubseteq \diamond^+ D$) we use a persistency (dynamic extension) constraint: $C \text{ PER } D$ ($C \text{ DEX } D$).
7. Axioms of the form $\text{next}(C, D)$ are mapped by using the dynamic extension constraints as showed in Figure 5(b).

The above reductions are enough to capture all axioms in \mathbf{KB}_M . We are now able to prove the first result of this paper.

Theorem 5.2. Reasoning over temporal class diagrams using persistency and dynamic constructs is undecidable.

Proof. We show that the mapping of \mathbf{KB}_M is correct. This will prove that the concept C_0 is satisfiable in \mathbf{KB}_M iff the class C_0 is satisfiable in Σ_M .

“ \Leftarrow ”. Let \mathcal{B} be a legal temporal database state for Σ_M , $\mathcal{B} = (\mathcal{T}, \Delta^{\mathcal{B}}, \cdot^{\mathcal{B}(t)})$, such that there exists $t_0 \in \mathcal{T} \cdot C_0^{\mathcal{B}(t_0)} \neq \emptyset$. We show that \mathcal{B} is a model for \mathbf{KB}_M , too. We proceed by induction on the structure of the axioms in \mathbf{KB}_M , after the elimination of conjunction, and disjunction between non-atomic concepts. Thus, we can just consider the following axioms where C, D, D_1, \dots, D_n are concept names.

⁶ Considering R as a functional role does not change the \mathcal{ALL}_F undecidability proof.

1. $\mathbf{C} \sqsubseteq \mathbf{D}$.

They are mapped in Σ_M as $C \text{ ISA } D$, thus, $\forall t \in \mathcal{T}. C^{\mathcal{B}(t)} \subseteq D^{\mathcal{B}(t)}$.

2. $\mathbf{C} \sqsubseteq \neg\mathbf{D}$.

They are mapped in Σ_M as in Figure 4(a), and, in particular, $\{C, D\} \text{ DISJ Top}$. Thus, for all $t \in \mathcal{T}$, $\text{Top}^{\mathcal{B}(t)} = \Delta^{\mathcal{B}}$, $C^{\mathcal{B}(t)} \subseteq \text{Top}^{\mathcal{B}(t)}$, $D^{\mathcal{B}(t)} \subseteq \text{Top}^{\mathcal{B}(t)}$, and, $C^{\mathcal{B}(t)} \cap D^{\mathcal{B}(t)} = \emptyset$. Then, $C^{\mathcal{B}(t)} \subseteq \Delta^{\mathcal{B}} \setminus D^{\mathcal{B}(t)}$.

3. $\text{discover}(\mathbf{C}, \mathbf{D}_1, \dots, \mathbf{D}_n)$.

They are mapped in Σ_M as:

$$\begin{aligned} \{D_1, \dots, D_n\} \text{ DISJ } C \\ \{D_1, \dots, D_n\} \text{ COVER } C \end{aligned}$$

Thus, for all $t \in \mathcal{T}$ and for all $i \in \{1, \dots, n\}$, then, $D_i^{\mathcal{B}(t)} \subseteq C^{\mathcal{B}(t)}$, $C^{\mathcal{B}(t)} = \bigcup_{i=1}^n D_i^{\mathcal{B}(t)}$, and, $\forall j \in \{1, \dots, n\}, j \neq i. D_i^{\mathcal{B}(t)} \cap D_j^{\mathcal{B}(t)} = \emptyset$.

4. $\mathbf{C} \sqsubseteq \mathbf{D}_1 \sqcup \dots \sqcup \mathbf{D}_n$.

They are mapped in Σ_M as in Figure 4(b), i.e., $\{D_1, \dots, D_n\} \text{ COVER } D$, and $C \text{ ISA } D$.

Thus, for all $t \in \mathcal{T}$, $D^{\mathcal{B}(t)} = \bigcup_{i=1}^n D_i^{\mathcal{B}(t)}$, and, $C^{\mathcal{B}(t)} \subseteq D^{\mathcal{B}(t)}$.

5. $\mathbf{C} \sqsubseteq \forall R.D$ (with R a functional global role).

They are mapped in Σ_M together with the axiom $\top \sqsubseteq \exists R.\top$ as in Figure 5(a). Then:

- $\text{REL}(R) = \langle U_1 : \text{Top}, U_2 : \text{Top} \rangle$, thus
 $\forall t \in \mathcal{T} \forall r \in R^{\mathcal{B}(t)}. r = \langle e_1, e_2 \rangle \in \Delta^{\mathcal{B}} \times \Delta^{\mathcal{B}}$
- $R \in \mathcal{R}^S$, thus
 $\forall t \in \mathcal{T}. r \in R^{\mathcal{B}(t)}$
- $\text{CARD}(\text{Top}, R, U_1) = (1, 1)$, thus
 $\forall e_1 \in \Delta^{\mathcal{B}}. \exists! e_2 \in \Delta^{\mathcal{B}}. \langle e_1, e_2 \rangle \in R^{\mathcal{B}(t)}$

Then, R is a functional and global role, and \mathcal{B} satisfies $\top \sqsubseteq \exists R.\top$. Furthermore:

- $\text{CARD}(C, U_1, R_c) = (1, 1)$, thus
 $\forall t \in \mathcal{T} \forall e_1 \in C^{\mathcal{B}(t)} \exists! e_2 \in \Delta^{\mathcal{B}}. \langle e_1, e_2 \rangle \in R_c^{\mathcal{B}(t)}$
- $\text{REL}(R_c) = \langle U_1 : C, U_2 : D \rangle$, thus
 $\forall t \in \mathcal{T} \forall \langle e_1, e_2 \rangle \in R_c^{\mathcal{B}(t)}. e_2 \in D^{\mathcal{B}(t)}$
- $R_c \text{ ISA } R$, thus
 $\forall t \in \mathcal{T} \forall \langle e_1, e_2 \rangle \in R_c^{\mathcal{B}(t)}. \langle e_1, e_2 \rangle \in R^{\mathcal{B}(t)}$

Then, since R is functional, \mathcal{B} satisfies $C \sqsubseteq \forall R.D$.

6. $\mathbf{C} \sqsubseteq \diamond^+ \mathbf{D}$ ($\mathbf{C} \sqsubseteq \square^+ \mathbf{D}$).

They are mapped in Σ_M as $C \text{ DEX } D$, thus, $\forall t \in \mathcal{T} \forall e \in C^{\mathcal{B}(t)} \exists t' > t. e \in D^{\mathcal{B}(t')}$. A similar proof holds for axioms of the form $\mathbf{C} \sqsubseteq \square^+ \mathbf{D}$.

7. $\text{next}(\mathbf{C}, \mathbf{D})$.

They are mapped in Σ_M as in Figure 5(b). In particular, since $C \text{ DEX } D$, then, by the previous point, \mathcal{B} is a model of $\mathbf{C} \sqsubseteq \diamond^+ \mathbf{D}$. Furthermore, since

$$\begin{array}{ll} C \text{ PER } C_1 & \text{then } \mathcal{B} \text{ is a model of } C \sqsubseteq \square^+ C_1 \\ C_1 \text{ PER } C_2 & \text{then } \mathcal{B} \text{ is a model of } C_1 \sqsubseteq \square^+ C_2 \\ \{C_2, D\} \text{ DISJ Top} & \text{then } \mathcal{B} \text{ is a model of } C_2 \sqsubseteq \neg D \end{array}$$

In summary, \mathcal{B} is a model of $C \sqsubseteq \diamond^+ D \sqcap \square^+ \square^+ \neg D$, i.e., \mathcal{B} is a model of $\text{next}(\mathbf{C}, \mathbf{D})$.

“ \Rightarrow ”. Let \mathcal{I} be a model for C_0 and \mathbf{KB}_M , $\mathcal{I} \doteq \langle \mathcal{T}, \Delta^{\mathcal{I}}, \mathcal{I}^{\mathcal{I}(t)} \rangle$. We construct a temporal interpretation $\mathcal{J} \doteq \langle \mathcal{T}, \Delta^{\mathcal{J}}, \mathcal{J}^{\mathcal{J}(t)} \rangle$ that is a model for Σ_M . \mathcal{J} gives the same interpretation as \mathcal{I} to all concepts and roles in (C_0, \mathbf{KB}_M) , and $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}}$ while $\text{Top}^{\mathcal{J}(t)} = \Delta^{\mathcal{I}}$, for all $t \in \mathcal{T}$. Furthermore, \mathcal{J} must interpret the additional classes and relationships introduced in Σ_M . We proceed by induction on the structure of the axioms in \mathbf{KB}_M by showing that

if \mathcal{I} is model of an axiom in \mathbf{KB}_M then \mathcal{J} satisfies the corresponding class diagram. The proof for axioms (1-4) is similar to the “ \Leftarrow ” direction.

5. $\mathbf{C} \sqsubseteq \forall \mathbf{R.D}$ (with R a functional global role).

They are mapped in Σ_M together with the axiom $\top \sqsubseteq \exists \mathbf{R.T}$ as in Figure 5(a). Then:

- Since \mathcal{I} models $\top \sqsubseteq \exists \mathbf{R.T}$, and R is a functional, total and global role, then, $\forall t \in \mathcal{T} \forall x \in \Delta^{\mathcal{I}} \exists! y \in \Delta^{\mathcal{I}}. (\langle x, y \rangle \in R^{\mathcal{I}(t)} \wedge \forall t' \in \mathcal{T}. \langle x, y \rangle \in R^{\mathcal{I}(t')})$. Since \mathcal{J} agrees with \mathcal{I} , and $\text{Top}^{\mathcal{J}(t)} = \Delta^{\mathcal{I}}$, it is easy to check that \mathcal{J} satisfies the portion of Figure 5(a) involving R and Top .
- Let us define $R_c^{\mathcal{J}(t)} = \{\langle x, y \rangle \in R^{\mathcal{I}(t)} \mid x \in C^{\mathcal{I}(t)}\}$. Since R is functional and total, and \mathcal{I} models $\mathbf{C} \sqsubseteq \forall \mathbf{R.D}$, then, $\forall x \in C^{\mathcal{I}(t)} \exists! y. (\langle x, y \rangle \in R^{\mathcal{I}(t)} \wedge y \in D^{\mathcal{I}(t)})$. Thus, by \mathcal{J} definition, $\forall x \in C^{\mathcal{J}(t)} \exists! y. (\langle x, y \rangle \in R_c^{\mathcal{J}(t)} \wedge y \in D^{\mathcal{J}(t)})$. In conclusion, \mathcal{J} satisfies Figure 5(a).

6. $\mathbf{C} \sqsubseteq \diamond^+ \mathbf{D}$ ($\mathbf{C} \sqsubseteq \square^+ \mathbf{D}$).

Similar to the “ \Leftarrow ” direction.

7. $\text{next}(\mathbf{C}, \mathbf{D})$.

They are mapped in Σ_M as in Figure 5(b). Let us define:

$$\begin{aligned} C_2^{\mathcal{J}(t)} &= (\neg D)^{\mathcal{J}(t)} \equiv (\neg D)^{\mathcal{I}(t)} \\ C_1^{\mathcal{J}(t)} &= (\square^+ C_2)^{\mathcal{J}(t)} \end{aligned}$$

Thus, \mathcal{J} satisfies the disjoint hierarchy involving C_2 and D , and the dynamic constraint $C_1 \text{ PER } C_2$. Since \mathcal{I} satisfies $\mathbf{C} \sqsubseteq \square^+ \square^+ \neg D$, and \mathcal{J} agrees with \mathcal{I} on C, D , then, \mathcal{J} satisfies $\mathbf{C} \sqsubseteq \square^+ \square^+ C_2$ and thus $\mathbf{C} \sqsubseteq \square^+ C_1$, i.e., \mathcal{J} satisfies $\mathbf{C} \text{ PER } C_1$. Finally, since \mathcal{I} satisfies $\mathbf{C} \sqsubseteq \diamond^+ D$, then, \mathcal{J} satisfies $\mathbf{C} \text{ DEX } D$.

□

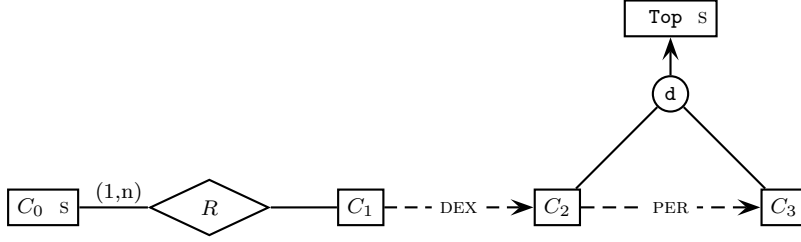
6. Finite Model Reasoning

An usual assumption in databases is that one of a finite universe. This Section shows that temporal class diagrams do not enjoy the *finite model property* (FMP, for short). This means that reasoning on finite models is different from reasoning on infinite ones as proved by the following theorem.

Theorem 6.1. Temporal class diagrams do not have the FMP.

Proof. Let us consider the schema, Σ_{inf} , of Figure 6. We show that C_0 is satisfiable only on models with infinite objects. Let \mathcal{B} be a model of Σ_{inf} such that $\exists e_0 \in \Delta^{\mathcal{B}}. e_0 \in C_0^{\mathcal{B}(t_0)}$, for some $t_0 \in \mathcal{T}$. Thus, $\exists e_1 \in \Delta^{\mathcal{B}}. (\langle e_0, e_1 \rangle \in R^{\mathcal{B}(t_0)} \wedge e_1 \in C_1^{\mathcal{B}(t_0)})$. Then, given the dynamic extension, $\exists t_1 > t_0. e_1 \in C_2^{\mathcal{B}(t_1)}$, while, given the persistency and the disjointness constraints, $\forall t > t_1. e_1 \notin C_2^{\mathcal{B}(t)}$, and then, $\forall t \geq t_1. e_1 \notin C_1^{\mathcal{B}(t)}$. Now, since C_0 is snapshot, $e_0 \in C_0^{\mathcal{B}(t)}$ for all $t \in \mathcal{T}$, and, in particular, $e_0 \in C_0^{\mathcal{B}(t_1)}$, while since C_0 totally participates in R , then, $\exists e'_1 \in \Delta^{\mathcal{B}}. (\langle e_0, e'_1 \rangle \in R^{\mathcal{B}(t_1)} \wedge e'_1 \in C_1^{\mathcal{B}(t_1)})$. Since $e'_1 \in C_1^{\mathcal{B}(t_1)}$, while $e_1 \notin C_1^{\mathcal{B}(t_1)}$, then, $e'_1 \neq e_1$. For similar reasons, $\exists t_2 > t_1$ such that $\forall t \geq t_2. e'_1 \notin C_1^{\mathcal{B}(t)}$. Thus, we need to introduce a new object, e''_1 to make C_0 satisfiable, and so on so forth. □

We are now able to show the second relevant result of this paper. The following theorem shows that the undecidability result also holds when reasoning w.r.t. finite models in temporal class diagrams.

Figure 6. The Σ_{inf} schema.

Theorem 6.2. Reasoning over temporal class diagrams using persistency and dynamic constructs is undecidable even when considering legal temporal databases with finite domains.

Proof. Obviously, also \mathcal{ALC}_F lacks the FMP. We then show that concept satisfiability w.r.t. an \mathcal{ALC}_F knowledge base is undecidable even considering finite models. The new \mathcal{ALC}_F axioms could be captured by temporal class diagrams by adopting the mapping already used in Theorem 5.2. Given a Turing machine, $\mathbf{M} = \langle A, S, \rho \rangle$, we construct an \mathcal{ALC}_F KB, say \mathbf{KB}_{fin} , with a concept that is satisfiable w.r.t. \mathbf{KB}_{fin} iff the machine \mathbf{M} does halt. The same notation introduced in Theorem 5.1 is used here. \mathbf{KB}_{fin} contains the following axioms:

$$C_0 \sqsubseteq C_{\mathcal{L}} \sqcap \diamond^+ C_{\langle s_0, b \rangle} \sqcap C_{halt} \quad (19)$$

$$\text{discover}(C, \{C_x \mid x \in A'\}) \quad (20)$$

$$C_{halt} \sqsubseteq \diamond^+ S1 \sqcup \exists R.C_{halt} \quad (21)$$

$$\text{next}(C_{\mathcal{L}}, D_1) \quad (22)$$

$$\text{next}(D_1, D_2) \quad (23)$$

$$C_{\langle s_0, b \rangle} \sqsubseteq D_1 \quad (24)$$

$$C_{\langle s_0, b \rangle} \sqsubseteq \square^+ C_b \quad (25)$$

$$\text{discover}(C_s, \{C_{\langle s, a \rangle} \mid \langle s, a \rangle \in S \times A\}) \quad (26)$$

$$\text{next}(C_l, C_s) \quad (27)$$

$$\text{next}(C_s, C_r) \quad (28)$$

$$\text{next}(C_r, D_3) \quad (29)$$

$$C_{\mathcal{L}} \sqsubseteq C_l \sqcup \diamond^+ C_l \quad (30)$$

$$C_l \sqsubseteq C_{\alpha} \rightarrow \forall R.C_{\alpha'} \quad (31)$$

$$C_s \sqsubseteq C_{\beta} \rightarrow \forall R.C_{\beta'} \quad (32)$$

$$C_r \sqsubseteq C_{\gamma} \rightarrow \forall R.C_{\gamma'} \quad (33)$$

$$C_a \sqsubseteq (\neg C_l \sqcap \neg C_s \sqcap \neg C_r) \rightarrow \forall R.C_a, \quad \forall a \in A \cup \{\mathcal{L}\} \quad (34)$$

$$\text{discover}(S1, \{C_{\langle s_1, a \rangle} \mid a \in A \cup \{\mathcal{L}\}\}) \quad (35)$$

$$C_{halt} \sqsubseteq \diamond^+ C_{chain} \quad (36)$$

$$\text{next}(C_{chain}, D_4) \quad (37)$$

$$D_4 \sqsubseteq \forall R.C_{chain} \quad (38)$$

with axioms (31–33) for each instruction $\delta(\alpha, \beta, \gamma) = \langle \alpha', \beta', \gamma' \rangle$. We now prove that C_0 is satisfiable w.r.t. \mathbf{KB}_{fin} iff \mathbf{M} has a finite computation starting from the empty tape.

“ \Rightarrow ” Let C_0 be satisfiable, then, $\exists \langle x_0, t_0 \rangle \in \Delta^{\mathcal{I}} \times \mathcal{T}. x_0 \in C_0^{\mathcal{I}(t_0)}$. Then, by axiom (19), $x_0 \in C_{halt}^{\mathcal{I}(t_0)}$, and, by axioms (36,37), $\exists t_1 > t_0. (x_0 \in C_{chain}^{\mathcal{I}(t_1)} \wedge x_0 \in D_4^{\mathcal{I}(t_1+1)})$. By axiom (21),

either $x_0 \in S1^{\mathcal{I}(t'_0)}$ for some $t'_0 > t_0$, or $\exists x_1 \in \Delta^{\mathcal{I}}. (\langle x_0, x_1 \rangle \in R^{\mathcal{I}(t_0)} \wedge x_1 \in C_{halt}^{\mathcal{I}(t_0)})$. If the last is true, we then show that $x_0 \neq x_1$. Indeed, since R is global, then, $\langle x_0, x_1 \rangle \in R^{\mathcal{I}(t_1+1)}$, and, by axiom (38), $x_1 \in C_{hain}^{\mathcal{I}(t_1+1)}$, and by axiom (37), $\forall t \neq t_1+1. x_1 \notin C_{hain}^{\mathcal{I}(t)}$. Since $x_0 \in C_{hain}^{\mathcal{I}(t_1)}$, then, $x_0 \neq x_1$. Thus, there is a chain of different objects in $\Delta^{\mathcal{I}}$, $\langle x_0, x_1, \dots, x_n \rangle$, with $n \geq 0$, such that $x_0 \in C_0^{\mathcal{I}(t_0)}$, and, $x_n \in S1^{\mathcal{I}(t'_n)}$, for some $t'_n > t_0$. The chain is finite since $\Delta^{\mathcal{I}}$ is finite. The fact that the chain $\langle x_0, x_1, \dots, x_n \rangle$ represents a computation of \mathbf{M} can be done similarly to Theorem 5.1.

“ \Leftarrow ” Conversely, suppose that \mathbf{M} is a Turing machine and $\langle c_0, \dots, c_n \rangle$ its finite computation starting with the empty tape. We construct a model $\mathcal{I} \doteq \langle \mathcal{T}, \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}(t)} \rangle$ of \mathbf{KB}_{fin} such that C_0 is satisfiable. In particular, we fix $\mathcal{T} = \langle \mathbb{N}, < \rangle$, $\Delta^{\mathcal{I}} = \{0, 1, \dots, n\} \subseteq \mathbb{N}$, $R^{\mathcal{I}} = \text{succ}_{\Delta^{\mathcal{I}}}$ (the successor function over \mathbb{N} restricted to $\Delta^{\mathcal{I}}$), $C_0^{\mathcal{I}(0)} = \{0\}$, $C_0^{\mathcal{I}(j)} = \emptyset$, for all $j > 0$, $C_{halt}^{\mathcal{I}(0)} = \Delta^{\mathcal{I}}$, and $C_{halt}^{\mathcal{I}(j)} = \emptyset$, for all $j > 0$. Furthermore, $\forall j \in \mathbb{N}$:

- $C_x^{\mathcal{I}(j)} = \{i \in \Delta^{\mathcal{I}} \mid \text{the } j\text{th cell of } c_i \text{ contains } x\}$, for all $x \in A'$
- $C_s^{\mathcal{I}(j)} = \{i \in \Delta^{\mathcal{I}} \mid \text{the } j\text{th cell of } c_i \text{ is the active one}\}$
- $C_l^{\mathcal{I}(j)} = C_s^{\mathcal{I}(j+1)}$
- $C_r^{\mathcal{I}(j)} = C_s^{\mathcal{I}(j-1)}$
- $C^{\mathcal{I}(j)} = \bigcup_{x \in A'} C_x^{\mathcal{I}(j)}$
- $C_{hain}^{\mathcal{I}(j)} = \begin{cases} \{j-1\} & \text{if } j \in \{1, \dots, n\} \\ \emptyset & \text{otherwise} \end{cases}$
- $D_1^{\mathcal{I}(j)} = C_{\ell}^{\mathcal{I}(j-1)}$
- $D_2^{\mathcal{I}(j)} = D_1^{\mathcal{I}(j-1)}$
- $D_3^{\mathcal{I}(j)} = C_r^{\mathcal{I}(j-1)}$
- $D_4^{\mathcal{I}(j)} = C_{hain}^{\mathcal{I}(j-1)}$
- $S1^{\mathcal{I}(j)} = \bigcup_{a \in A} C_{\langle s_1, a \rangle}^{\mathcal{I}(j)}$.

It is easy to verify that \mathcal{I} is a model of \mathbf{KB}_{fin} where C_0 is satisfiable. \square

7. Conclusions

We formally introduced a data modeling language useful to represent time-varying data. The language is equipped with a linear and graphical syntax and a model-theoretic semantics. A relevant aspect of the proposed formalism is the possibility to formally specify reasoning tasks based on the associated semantics. Reasoning problems as class, relationship and schema satisfiability and logical implication have been described.

We then investigated the complexity of reasoning on temporal models and we found that such problem is undecidable as soon as the language is able to distinguish between temporal and atemporal constructs (in particular, whether the language captures temporal relationships) and has the ability to represent dynamic constraints between classes. While temporal class diagrams do not enjoy the finite model property we prove that even reasoning on finite models is undecidable.

The main reason behind the undecidability result is the possibility to postulate that a binary relation does not vary in time. Indeed, it has been shown in [5] that temporal diagrams expressed in the \mathcal{ER}_{VT} modeling language can be embedded into the temporal description logic \mathcal{DLR}_{US} . While \mathcal{DLR}_{US} is undecidable, the fragment, \mathcal{DLR}_{US}^- , of \mathcal{DLR}_{US} deprived of the ability to talk about temporal persistence of n -ary relations, for $n \geq 2$, is decidable. Indeed, reasoning in \mathcal{DLR}_{US}^- is an EXPTIME-complete problem [5].

This result gives us an useful scenario where reasoning over temporal schemas becomes decidable. In particular, if we forbid timestamping for relationships (i.e., relationships are just unmarked) reasoning on temporal models with both concept timestamping and full evolution constraints can be reduced to reasoning over \mathcal{DLR}_{US}^- . The problem of reasoning in this setting is complete for EXPTIME since the EXPTIME-complete problem of reasoning with \mathcal{ALC} knowledge bases can be captured by such schemas [7].

It is an open problem whether reasoning is still decidable by regaining timestamping for relationships (and maintaining timestamping for classes) but dropping evolution constraints altogether. We have a strong feeling that this represents a decidable scenario since it is possible to encode temporal schemas without evolution constraints by using a combination between the description logic \mathcal{DLR} and the epistemic modal logic $S5$. Decidability results have been proved for the sub-logic \mathcal{ALC}_{S5} [12]. But, it is an open problem whether this result still holds for the more complex logic \mathcal{DLR}_{S5} .

Acknowledgments

The author would like to thank Diego Calvanese, Enrico Franconi, Sergio Tessaris and Frank Wolter for enlightening comments on earlier drafts of the paper. The author has been partially supported by the EU projects KnowledgeWeb, Interop and Tones.

References

- [1] A. Artale and E. Franconi. Temporal ER modeling with description logics. In *Proc. of the International Conference on Conceptual Modeling (ER'99)*. Lecture Notes in Computer Science, Springer-Verlag, 1999.
- [2] A. Artale and E. Franconi. A survey of temporal extensions of description logics. *Annals of Mathematics and Artificial Intelligence*, 30(1-4), 2001.
- [3] Alessandro Artale. Reasoning on temporal conceptual schemas with dynamic constraints. In *11th International Symposium on Temporal Representation and Reasoning (TIME04)*. IEEE Computer Society, 2004. Also in Proc. of the 2004 International Workshop on Description Logics (DL'04).
- [4] Alessandro Artale, Enrico Franconi, and Federica Mandreoli. Description logics for modelling dynamic information. In Jan Chomicki, Ron van der Meyden, and Gunter Saake, editors, *Logics for Emerging Applications of Databases*. Lecture Notes in Computer Science, Springer-Verlag, 2003.
- [5] Alessandro Artale, Enrico Franconi, Frank Wolter, and Michael Zakharyashev. A temporal description logic for reasoning about conceptual schemas and queries. In S. Flesca, S. Greco, N. Leone, and G. Ianni, editors, *Proceedings of the 8th Joint European Conference on Logics in Artificial Intelligence (JELIA-02)*, volume 2424 of *LNAI*, pages 98–110. Springer, 2002.
- [6] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2002.
- [7] Daniela Berardi, Diego Calvanese, and Giuseppe De Giacomo. Reasoning on UML class diagrams. *Artificial Intelligence*, 168(1–2):70–118, 2005.
- [8] 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.
- [9] D. Calvanese, M. Lenzerini, and D. Nardi. Unifying class-based representation formalisms. *J. of Artificial Intelligence Research*, 11:199–240, 1999.
- [10] J. Chomicki and D. Toman. Temporal logic in information systems. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*, chapter 1. Kluwer, 1998.
- [11] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems*. Benjamin/Cummings, 2nd edition, 1994.
- [12] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyashev. *Many-dimensional modal logics: theory and applications*. Studies in Logic. Elsevier, 2003.
- [13] H. Gregersen and J.S. Jensen. Conceptual modeling of time-varying information. Technical Report TimeCenter TR-35, Aalborg University, Denmark, 1998.
- [14] H. Gregersen and J.S. Jensen. Temporal Entity-Relationship models - a survey. *IEEE Transactions on Knowledge and Data Engineering*, 11(3):464–497, 1999.
- [15] R. Gupta and G. Hall. Modeling transition. In *Proc. of ICDE'91*, pages 540–549, 1991.

- [16] 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.
- [17] C. S. Jensen and R. T. Snodgrass. Temporal data management. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):36–44, 1999.
- [18] C. S. Jensen, M. Soo, and R. T. Snodgrass. Unifying temporal data models via a conceptual model. *Information Systems*, 9(7):513–547, 1994.
- [19] S. Spaccapietra, C. Parent, and E. Zimanyi. Modeling time from a conceptual perspective. In *Int. Conf. on Information and Knowledge Management (CIKM98)*, 1998.
- [20] C. Theodoulidis, P. Loucopoulos, and B. Wangler. A conceptual modelling formalism for temporal database applications. *Information Systems*, 16(3):401–416, 1991.
- [21] F. Wolter and M. Zakharyashev. Satisfiability problem in description logics with modal operators. In *Proc. of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 512–523, Trento, Italy, June 1998.