Reasoning on Temporal Conceptual Schemas with Dynamic Constraints

Alessandro Artale*
Faculty of Computer Science – Free University of Bozen-Bolzano
artale@inf.unibz.it

1 Introduction

Temporally enhanced conceptual models have been developed to help designing temporal databases [12]. In this paper we deal with Extended Entity-Relationship (EER) diagrams[1] used to model temporal databases. The temporal conceptual model $\mathcal{ER}_{VT}$ has been introduced both to *formally* clarify the meaning of the various temporal constructs appeared in the literature [2, 4], and to check the possibility to perform *reasoning* on top of temporal schemas [5]. $\mathcal{ER}_{VT}$ supports valid time for entities, attributes, and relationships in the line of TIMEER [10] and ERT [15], while supporting dynamic constraints for entities as presented in MADS [14]. $\mathcal{ER}_{VT}$ is able to distinguish 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 capture the *object migration* from a source entity to a target entity.

The contribution of this paper is twofold. Moving from the formal characterization of $\mathcal{ER}_{VT}$ given in [4] we clarify the relevant reasoning problems for temporal EER diagrams. In particular, we distinguish between six different reasoning services, introducing two new services for both entities and relationships: *liveness satisfiability*—i.e. whether an entity or relationship admits a non-empty extension infinitely often in the future—and *global satisfiability*—i.e. whether an entity or relationship admits a non-empty extension at all points in time. After a systematic definition of the various reasoning problems we then show that all the satisfiability problems (i.e. schema, entity and relationship satisfiability problems) together with the subsumption problem (i.e. checking whether two entities or relationships denote one a subset of the other so that there is an implicit ISA link between them) can be mutually reduced to each other. On the other hand, checking whether a schema *logically implies* another schema is shown to be the more general reasoning service.

The second contribution is to prove that reasoning on temporal conceptual models is undecidable provided the diagrams are able to: (a) Distinguish between temporal and non-temporal constructs; (b) Represent *dynamic constraints* between entities, i.e. entities whose instances

*The author has been partially supported by the EU projects Sewasie, KnowledgeWeb, and Interop. This paper is a shorter version of [1].

[1]EER is the standard entity-relationship data model, enriched with ISA links, generalized hierarchies with disjoint and covering constraints, and full cardinality constraints.

$$
\begin{array}{llllrl}
C, D & \rightarrow & A \mid & \text{(atomic concept)} & A^{\mathcal{I}(t)} \subseteq & \Delta^{\mathcal{I}} \\
 & & \top \mid & \text{(top)} & \top^{\mathcal{I}(t)} = & \Delta^{\mathcal{I}} \\
 & & \bot \mid & \text{(bottom)} & \bot^{\mathcal{I}(t)} = & \emptyset \\
 & & \neg C \mid & \text{(complement)} & (\neg C)^{\mathcal{I}(t)} = & \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}(t)} \\
 & & C \sqcap D \mid & \text{(conjunction)} & (C \sqcap D)^{\mathcal{I}(t)} = & C^{\mathcal{I}(t)} \cap D^{\mathcal{I}(t)} \\
 & & C \sqcup D \mid & \text{(disjunction)} & (C \sqcup D)^{\mathcal{I}(t)} = & C^{\mathcal{I}(t)} \cup D^{\mathcal{I}(t)} \\
 & & \exists R.C \mid & \text{(exist. quantifier)} & (\forall R.C)^{\mathcal{I}(t)} = & \{a \in \Delta^{\mathcal{I}} \mid \forall b. R^{\mathcal{I}(t)}(a,b) \Rightarrow C^{\mathcal{I}(t)}(b)\} \\
 & & \forall R.C \mid & \text{(univ. quantifier)} & (\exists R.C)^{\mathcal{I}(t)} = & \{a \in \Delta^{\mathcal{I}} \mid \exists b. R^{\mathcal{I}(t)}(a,b) \wedge C^{\mathcal{I}(t)}(b)\} \\
 & & \Diamond^+ C \mid & \text{(Sometime)} & (\Diamond^+ C)^{\mathcal{I}(t)} = & \{a \in \Delta^{\mathcal{I}} \mid \exists v > t. C^{\mathcal{I}(v)}(a)\} \\
 & & \Box^+ C \mid & \text{(Every time)} & (\Box^+ C)^{\mathcal{I}(t)} = & \{a \in \Delta^{\mathcal{I}} \mid \forall v > t. C^{\mathcal{I}(v)}(a)\}
\end{array}
$$

Figure 1: Syntax and Semantics for the $\mathcal{ALC}_{\mathsf{F}}$ Description Logic

migrate to other entities. To the best of our knowledge, this is the first time such a result is proved. Indeed, the result presented in [5] showed that $\mathcal{ER}_{VT}$ diagrams can be embedded into the temporal description logic (DL) $\mathcal{DLR}_{\mathcal{US}}$—where $\mathcal{U}$, $\mathcal{S}$ extend $\mathcal{DLR}$ with the *until* and *since* temporal modalities—and that reasoning in $\mathcal{DLR}_{\mathcal{US}}$ was undecidable. Instead, here we prove that even reasoning just on $\mathcal{ER}_{VT}$ schemas is undecidable. The undecidability result is proved via a reduction of the Halting Problem with a technique similar to [9]. In particular, we proceed by first showing that the halting problem can be encoded as a Knowledge Base (KB) in $\mathcal{ALC}_{\mathsf{F}}$—where F extends $\mathcal{ALC}$ with the *future* temporal modality—and then proving that such a KB in $\mathcal{ALC}_{\mathsf{F}}$ can be captured by an $\mathcal{ER}_{VT}$ diagram.

The paper is organized as follows. The temporal DL $\mathcal{ALC}_{\mathsf{F}}$ and the conceptual model $\mathcal{ER}_{VT}$ are formally presented in Sections 2 and 3, respectively. The various reasoning services for temporal conceptual modeling are defined in Section 4 and their equivalence is proved. That reasoning in presence of dynamic constraints is undecidable is proved in Section 5.

2 The Temporal Description Logic

In this Section we introduce the $\mathcal{ALC}_{\mathsf{F}}$ DL [16, 3, 9] as a the tense-logical extension of $\mathcal{ALC}$. Basic types of $\mathcal{ALC}_{\mathsf{F}}$ are *concepts* and *roles*. According to the syntax rules of Figure 1, $\mathcal{ALC}_{\mathsf{F}}$ *concepts* are built out of *atomic concepts* and *atomic roles*. Tense operators are added for concepts: $\Diamond^+$ (sometime in the future) and $\Box^+$ (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 $\mathcal{ALC}_{\mathsf{F}}$. 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 $\mathcal{ALC}_{\mathsf{F}}$ *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 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 $\mathcal{ALC}_{\mathsf{F}}$ concepts is defined in Figure 1.

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 $C \sqsubseteq D$ iff 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 $t \in \mathcal{T}$. A knowledge base Σ is *satisfiable* if there is a temporal interpretation $\mathcal{I}$ that satisfies every axiom in 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. 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. $\Sigma \not\models C \sqsubseteq \bot$.

3 Temporal Conceptual Modeling

In this Section, the temporal EER model $\mathcal{ER}_{VT}$ is briefly introduced. $\mathcal{ER}_{VT}$ supports valid time for entities, attributes, and relationships in the line of TIMEER [10] and ERT [15], while supporting dynamic constraints for entities as presented in MADS [14]. $\mathcal{ER}_{VT}$ is able to distinguish between *snapshot* (see the consensus glossary [11] for the terminology used) constructs—i.e. each of their instances has a global lifespan—*temporary* constructs—i.e. each of their instances have a limited lifespan—or *implicitly temporal* constructs—i.e. their instances can have either a global or a temporary existence. Two temporal marks, S (snapshot) and VT (valid time), are introduced in $\mathcal{ER}_{VT}$ to capture such temporal behavior.

Dynamic constructs capture the *object migration* from a source entity to a target entity. If there is a *dynamic extension* between a source and a target entity (represented in $\mathcal{ER}_{VT}$ by a dotted link labeled with DEX) models the case where instances of the source entity *eventually* become instances of the target entity. On the other hand, a *dynamic persistency* (represented in $\mathcal{ER}_{VT}$ by a dotted link labeled with PER) models the dual case of instances *persistently* migrating to a target entity (for a complete introduction on $\mathcal{ER}_{VT}$ with a worked out example see [4]).

$\mathcal{ER}_{VT}$ is equipped with both a linear and a graphical syntax along with a model-theoretic semantics as a temporal extension of the EER semantics [7]. Presenting the $\mathcal{ER}_{VT}$ 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 $\{x_1, \ldots, x_n\} \subseteq X$ to the set $\{y_1, \ldots, y_n\} \subseteq Y$ is denoted by $\langle x_1 : y_1, \ldots, x_n : y_n \rangle$, and $T[x_i] = y_i$. An $\mathcal{ER}_{VT}$ schema is a tuple:

$$\Sigma = (\mathcal{L}, \text{REL}, \text{ATT}, \text{CARD}, \text{ISA}, \text{DISJ}, \text{COVER}, \text{S}, \text{T}, \text{KEY}, \text{DEX}, \text{PER}), \text{ such that:}$$

$\mathcal{L}$ is a finite alphabet partitioned into the sets: $\mathcal{E}$ (*entity* symbols), $\mathcal{A}$ (*attribute* symbols), $\mathcal{R}$ (*relationship* symbols), $\mathcal{U}$ (*role* symbols), and $\mathcal{D}$ (*domain* symbols). $\mathcal{E}$ is further partitioned into: a set $\mathcal{E}^S$ of *snapshot entities* (the S-*marked* entities in Figure 2), a set $\mathcal{E}^I$ of I*mplicitly temporal entities* (the *unmarked* entities in Figure 2), and a set $\mathcal{E}^T$ of *temporary entities* (the VT-*marked* entities in Figure 2). A similar partition applies to the set $\mathcal{R}$. ATT is a function that maps an entity symbol in $\mathcal{E}$ to an $\mathcal{A}$-labeled tuple over $\mathcal{D}$, $\text{ATT}(E) = \langle A_1 : D_1, \ldots, A_h : D_h \rangle$. REL is a function that maps a relationship symbol in $\mathcal{R}$ to an $\mathcal{U}$-labeled tuple over $\mathcal{E}$, $\text{REL}(R) = \langle U_1 : E_1, \ldots, U_k : E_k \rangle$, and k is the *arity* of R. CARD is a function $\mathcal{E} \times \mathcal{R} \times \mathcal{U} \mapsto \mathbb{N} \times (\mathbb{N} \cup \{\infty\})$ denoting cardinality constraints. We denote with $\text{CMIN}(E, R, U)$ and $\text{CMAX}(E, R, U)$ the first and second component of CARD. In Figure 2, $\text{CARD}(\texttt{TopManager}, \texttt{Manages}, \texttt{man}) = (1, 1)$. ISA is a binary relationship $\text{ISA} \subseteq (\mathcal{E} \times \mathcal{E}) \cup (\mathcal{R} \times \mathcal{R})$. ISA between relationships is restricted to relationships with the same

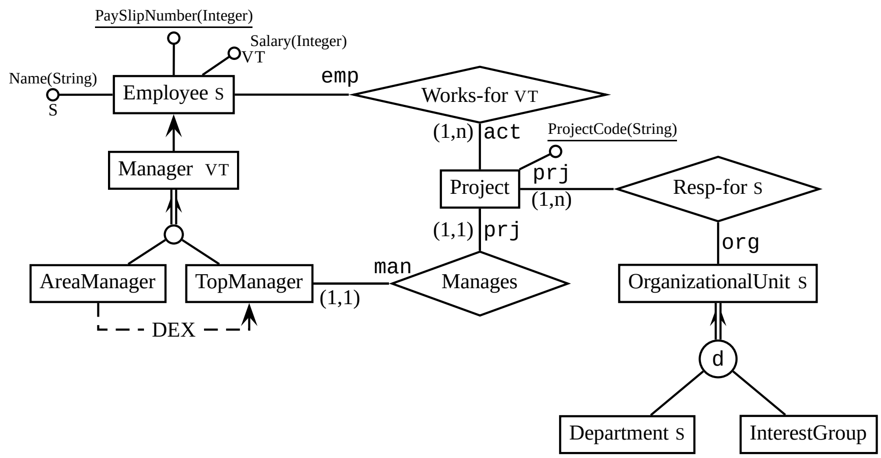

Figure 2: An $\mathcal{ER}_{VT}$ diagram

arity. ISA is visualized with a directed arrow, e.g. `Manager` ISA `Employee` in Figure 2. DISJ, COVER are binary relations over $2^{\mathcal{E}} \times \mathcal{E}$, describing disjointness and covering partitions, respectively. DISJ is visualized with a circled "d" and COVER with a double directed arrow, e.g. `Department`, `InterestGroup` are both disjoint and they cover `OrganizationalUnit`. S, T are binary relations over $\mathcal{E} \times \mathcal{A}$ containing, respectively, the snapshot and temporary attributes of an entity (see S, T marked attributes in Figure 2). KEY is a function that maps entity symbols in $\mathcal{E}$ to their key attributes, $\text{KEY}(E) = A$. Keys are visualized as underlined attributes. Both DEX and PER are binary relations over $\mathcal{E} \times \mathcal{E}$ describing the dynamic evolution of entities. DEX and PER are visualized with dotted directed lines labeled with DEX or PER, respectively (e.g. `AreaManager` DEX `TopManager`).

The model-theoretic semantics associated with the $\mathcal{ER}_{VT}$ modeling language adopts the *snapshot*[2] representation of abstract temporal databases and temporal conceptual models [8]. 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 either $\langle \mathbb{Z}, < \rangle$ 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 3.1 ($\mathcal{ER}_{VT}$ Semantics). *Let Σ be an $\mathcal{ER}_{VT}$ schema. A* temporal database state *for the schema Σ is a tuple $\mathcal{B} = (\mathcal{T}, \Delta^{\mathcal{B}} \cup \Delta^{\mathcal{B}}_D, \cdot^{\mathcal{B}(t)})$, such that: $\Delta^{\mathcal{B}}$ is a nonempty set disjoint from $\Delta^{\mathcal{B}}_D$; $\Delta^{\mathcal{B}}_D = \bigcup_{D_i \in \mathcal{D}} \Delta^{\mathcal{B}}_{D_i}$ is the set of basic domain values used in the schema Σ; $\cdot^{\mathcal{B}(t)}$ is a function such that for each $t \in \mathcal{T}$, every domain symbol $D_i \in \mathcal{D}$, every entity $E \in \mathcal{E}$, every relationship $R \in \mathcal{R}$, and every attribute $A \in \mathcal{A}$, we have: $D_i^{\mathcal{B}(t)} = \Delta^{\mathcal{B}}_{D_i}$, $E^{\mathcal{B}(t)} \subseteq \Delta^{\mathcal{B}}$, $R^{\mathcal{B}(t)}$ is a set of $\mathcal{U}$-labeled tuples over $\Delta^{\mathcal{B}}$, and $A^{\mathcal{B}(t)} \subseteq \Delta^{\mathcal{B}} \times \Delta^{\mathcal{B}}_D$. $\mathcal{B}$ is a* legal temporal database state *if it satisfies all integrity constraints expressed in the schema. In particular, the interpretation of* ISA, ATT, REL, CARD, DISJ.COVER *is similar to the atemporal case (see [7, 4]). For the temporal constructs we have:*

[2] The snapshot model represents the same class of temporal databases as the *timestamp* model [12, 13] defined by adding temporal attributes to a relation [8].

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

For each temporary entity $E \in \mathcal{E}^T$*, if,* $e \in E^{\mathcal{B}(t)}$*, then,* $\exists t' \neq t. e \notin E^{\mathcal{B}(t')}$.

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

For each temporary relationship $R \in \mathcal{R}^T$*, if,* $r \in R^{\mathcal{B}(t)}$*, then,* $\exists t' \neq t. r \notin R^{\mathcal{B}(t')}$.

For each entity $E \in \mathcal{E}$ *with a snapshot attribute* A_i*, i.e.* $\langle E, A_i \rangle \in \textsc{s}$*, if,* $(e \in E^{\mathcal{B}(t)} \wedge \langle e, a_i \rangle \in A_i^{\mathcal{B}(t)})$*, then,* $\forall t' \in \mathcal{T}. \langle e, a_i \rangle \in A_i^{\mathcal{B}(t')}$.

For each entity $E \in \mathcal{E}$ *with a temporary attribute* A_i*, i.e.* $\langle E, A_i \rangle \in \textsc{t}$*, if,* $(e \in E^{\mathcal{B}(t)} \wedge \langle e, a_i \rangle \in A_i^{\mathcal{B}(t)})$*, then,* $\exists t' \neq t. \langle e, a_i \rangle \notin A_i^{\mathcal{B}(t')}$.

For each $E \in \mathcal{E}, A \in \mathcal{A}$ *such that* $\textsc{key}(E) = A$*, then,* $\langle E, A_i \rangle \in \textsc{s}$*—i.e. a key is a snapshot attribute—and* $\forall a \in \Delta_D^{\mathcal{B}}. \#\{e \in E^{\mathcal{B}(t)} \mid \langle e, a \rangle \in A^{\mathcal{B}(t)}\} \leq 1$.

For each $E_1, E_2 \in \mathcal{E}$*, if* E_1 DEX E_2*, if,* $e \in E_1^{\mathcal{B}(t)}$*, then,* $\exists t_1 > t. e \in E_2^{\mathcal{B}(t_1)}$*;*

For each $E_1, E_2 \in \mathcal{E}$*, if* E_1 PER E_2*, if,* $e \in E_1^{\mathcal{B}(t)}$*, then,* $\forall t' > t. e \in E_2^{\mathcal{B}(t')}$.

4 Reasoning on Temporal Models

Reasoning tasks over a temporal conceptual model include verifying whether an entity, relationship, or schema are *satisfiable*, whether a *subsumption* relation exists between entities or relationships, or checking whether a new schema property is *logically implied* by a given schema. The model-theoretic semantics associated with $\mathcal{ER}_{VT}$ allows us to formally define these reasoning tasks.

Definition 4.1 (Reasoning in $\mathcal{ER}_{VT}$). *Let* Σ *be an* $\mathcal{ER}_{VT}$ *schema,* $E \in \mathcal{E}$ *an entity, and* $R \in \mathcal{R}$ *a relationship. The following are the reasoning tasks over* Σ*:*

1. E *(*R*) is* satisfiable *if there exists a legal temporal database state* $\mathcal{B}$ *for* Σ *such that* $E^{\mathcal{B}(t)} \neq \emptyset$ *(*$R^{\mathcal{B}(t)} \neq \emptyset$*), for some* $t \in \mathcal{T}$*;*

2. E *(*R*) is* liveness satisfiable *if there exists a legal temporal database state* $\mathcal{B}$ *for* Σ *such that* $\forall t \in \mathcal{T}. \exists t' > t. E^{\mathcal{B}(t')} \neq \emptyset$ *(*$R^{\mathcal{B}(t')} \neq \emptyset$*), i.e.* E *(*R*) is satisfiable infinitely often;*

3. E *(*R*) is* globally satisfiable *if there exists a legal temporal database state* $\mathcal{B}$ *for* Σ *such that* $E^{\mathcal{B}(t)} \neq \emptyset$ *(*$R^{\mathcal{B}(t)} \neq \emptyset$*), for all* $t \in \mathcal{T}$*;*

4. Σ *is* satisfiable *if there exists a legal temporal database state* $\mathcal{B}$ *for* Σ *that satisfies at least one entity in* Σ *(*$\mathcal{B}$ *is said a* model *for* Σ*);*

5. E_1 *(*R_1*) is* subsumed *by* E_2 *(*R_2*) in* Σ *if every legal temporal database state for* Σ *is also a legal temporal database state for* E_1 ISA E_2 *(*R_1 ISA R_2*);*

6. *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'.

Based on this formal characterization the following Proposition proves that reasoning services (1-5) relative to entities are mutually reducible to each other. As far as relationships are concerned, the reasoning services (1-3) can be reduced to analogous problems for entities.

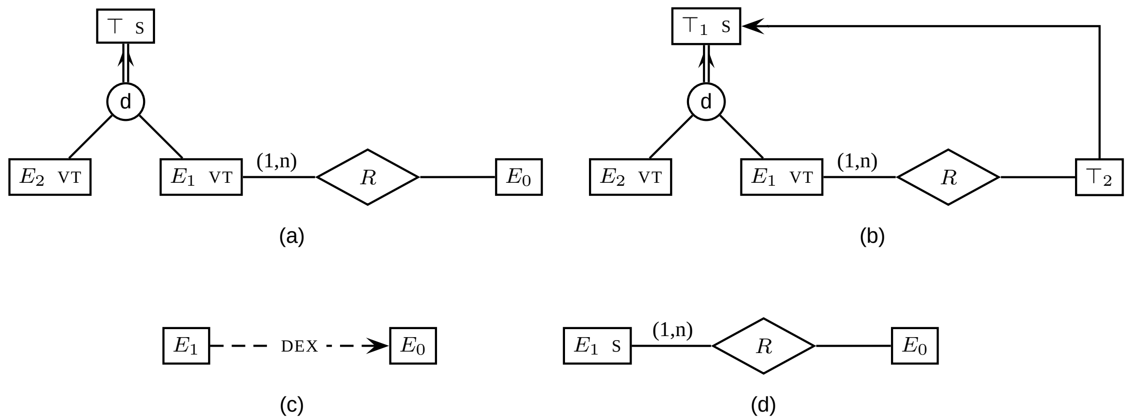

Figure 3: Reductions: (a) From Entity Sat to Schema Sat; (b) From Schema Sat to Entity Liveness Sat; (c) From Entity Liveness Sat to Entity Global Sat; (d) From Entity Global Sat to Entity Sat.

Indeed, we can verify whether a relationship R is satisfiable in Σ by adding a new entity, say A_R such that: (a) A_R ISA E, with E an arbitrary entity participating in the relationship, and (b) A_R totally participates in the relationship. Then, R is satisfiable (liveness or globally satisfiable) if and only if A_R is satisfiable (liveness or globally satisfiable). As far as relationships subsumption is concerned, it can be reduced to relationships satisfiability by extending $\mathcal{ER}_{VT}$ to express disjoint hierarchies between relationships and then applying the reduction proposed by [6] for entities.

Proposition 4.2. *There is a mutual reducibility between the reasoning services (1-5) in $\mathcal{ER}_{VT}$.*

Proof. (Sketch.)

1. Proving the mutual reducibility between satisfiability and subsumption in $\mathcal{ER}_{VT}$ can be done similarly to [6].

2. Entity satisfiability reduces to schema satisfiability.
   An arbitrary entity, E_0, is satisfiable w.r.t. Σ iff a new schema Σ' is satisfiable. Σ' is obtained by adding to Σ the schema in Figure 3(a), where $\top, E_1, E_2$ are new entities such that $\forall E \in \mathcal{E}.E$ ISA $\top$, and R is a new binary relationship.

3. Schema satisfiability reduces to entity liveness satisfiability.
   An arbitrary schema Σ is satisfiable iff an entity is liveness satisfiable w.r.t. a new schema Σ'. Σ' is obtained by adding to Σ the schema in Figure 3(b), where $\top_1, \top_2, E_1, E_2$ are new entities and R is a new binary relationship. Furthermore, $\{E \mid E \in \mathcal{E}\}$ COVER $\top_2$. In particular, Σ is satisfiable iff $\top_1$ is liveness satisfiable w.r.t. Σ'.

4. Entity liveness satisfiability reduces to entity global satisfiability.
   An arbitrary entity, E_0, is liveness satisfiable w.r.t. Σ iff an entity is globally satisfiable w.r.t. a new schema Σ'. Σ' is obtained by adding to Σ the new entity E_1 as shown in Figure 3(c). In particular, E_0 is liveness satisfiable w.r.t. Σ iff E_1 is globally satisfiable w.r.t. Σ'.

5. Entity global satisfiability reduces to entity satisfiability.
   An arbitrary entity, E_0, is globally satisfiable w.r.t. Σ iff the new entity E_1 is satisfiable w.r.t. the new schema Σ'. Σ' is obtained by adding to Σ the schema in Figure 3(d), where E_1 is new snapshot entity and R is a new binary relationship. $\square$

Finally, we show that all the reasoning problems can be reduced to a logical implication problem. Indeed, checking whether an entity E is satisfiable can be reduced to logical implication by choosing $\Sigma' = \{E \text{ ISA } A, E \text{ ISA } B, \{A, B\} \text{ DISJ } C\}$, with A, B, C arbitrary entities. Then, E is satisfiable iff $\Sigma \not\models \Sigma'$. Given the result of Proposition 4.2, then the reasoning services (1-5) for entities are reducible to logical implication. Furthermore, given two relationships R_1, R_2, checking for sub-relationship can be reduced to logical implication by choosing $\Sigma' = \{R_1 \text{ ISA } R_2\}$.

5 Reasoning on $\mathcal{ER}_{VT}$ is Undecidable

We now show that reasoning on full $\mathcal{ER}_{VT}$ is undecidable. The proof is based on a reduction from the undecidable halting problem for a Turing machine to the entity satisfiability problem w.r.t. an $\mathcal{ER}_{VT}$ schema Σ. We apply ideas similar to [9] (Sect. 7.5) to show undecidability of certain products of modal logics. The proof can be divided in the following two steps: 1. Reduction of the halting problem to concept satisfiability w.r.t. an $\mathcal{ALC}_\mathsf{F}$ KB; 2. Reduction of concept satisfiability w.r.t. an $\mathcal{ALC}_\mathsf{F}$ KB to entity satisfiability w.r.t. an $\mathcal{ER}_{VT}$ schema.

Reasoning on $\mathcal{ALC}_\mathsf{F}$ is undecidable

Using a reduction from the halting problem we now prove that reasoning involving an $\mathcal{ALC}_\mathsf{F}$ knowledge base is undecidable. In [9] the undecidability of $\mathcal{ALC}_\mathsf{F}$ 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 $\mathcal{ER}_{VT}$ is able to encode just simple global axioms, we modify the proof presented in [9].

Proposition 5.1. *Concept satisfiability w.r.t. an $\mathcal{ALC}_\mathsf{F}$ KB is undecidable.*

Proof. (`Sketch.`) 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 *initial state*, s_0, and *final state*, s_1; ρ is the *transition function*, $\rho : (S - \{s_1\}) \times A \to S \times (A \cup \{\mathtt{L}, \mathtt{R}\})$. We construct an $\mathcal{ALC}_\mathsf{F}$ 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 introduce some shortcuts. The implication, $C \to D$, is equivalent to $\neg C \sqcup D$. We define $\mathtt{next}(C, D)$ as: $C \sqsubseteq \Diamond^+ D \sqcap \neg \Diamond^+ \Diamond^+ D$. Finally, $\mathtt{discover}(C, \{D_1, \ldots, D_n\})$ is the disjoint covering between C and $D_1 \ldots D_n$. Let $A' = A \cup \{\pounds\} \cup (S \times A)$, where $\pounds \notin A$ is a symbol marking the left end of the tape. 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:

$$C_0 \sqsubseteq C_{\pounds} \sqcap \Diamond^+ C_{\langle s_0,b\rangle} \quad (1)$$
$$\top \sqsubseteq \exists R.\top \quad (2)$$
$$\mathtt{next}(C_{\pounds}, D_1) \quad (3)$$
$$\mathtt{next}(D_1, D_2) \quad (4)$$
$$C_{\langle s_0,b\rangle} \sqsubseteq D_1 \quad (5)$$
$$C_{\langle s_0,b\rangle} \sqsubseteq \Box^+ C_b \quad (6)$$
$$\mathtt{next}(C_l, C_s) \quad (7)$$
$$\mathtt{next}(C_s, C_r) \quad (8)$$
$$\mathtt{next}(C_r, D_3) \quad (9)$$
$$C_{\pounds} \sqsubseteq C_l \sqcup \Diamond^+ C_l \quad (10)$$
$$C_l \sqsubseteq C_\alpha \to \forall R.C_{\alpha'} \quad (11)$$
$$C_s \sqsubseteq C_\beta \to \forall R.C_{\beta'} \quad (12)$$
$$C_r \sqsubseteq C_\gamma \to \forall R.C_{\gamma'} \quad (13)$$
$$C_a \sqsubseteq (\neg C_l \sqcap \neg C_s \sqcap \neg C_r) \to \forall R.C_a, \ \ \forall a \in A \cup \{\pounds\} \quad (14)$$
$$\mathtt{discover}(S1, \{C_{\langle s_1,a\rangle} \mid a \in A \cup \{\pounds\}\}) \quad (15)$$
$$\mathtt{discover}(C, \{C_x \mid x \in A'\}) \quad (16)$$
$$\mathtt{discover}(C_s, \{C_{\langle s,a\rangle} \mid \langle s,a\rangle \in S \times A\}) \quad (17)$$
$$C_s \sqsubseteq \neg S1 \quad (18)$$

with axioms (11–13) for each instruction, $\delta(\alpha, \beta, \gamma) = \langle \alpha', \beta', \gamma' \rangle$, defined as

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

We can prove that C_0 is satisfiable w.r.t. $\mathbf{KB}_M$ iff $\mathbf{M}$ has an infinite computation starting from the empty tape. □

Reducing $\mathcal{ALC}_\mathsf{F}$ concept sat to $\mathcal{ER}_{VT}$ entity sat

We now show how to capture the $\mathcal{ALC}_\mathsf{F}$ knowledge base $\mathbf{KB}_M$ with an $\mathcal{ER}_{VT}$ schema, Σ_M. The mapping is based on a similar reduction presented in [6] for capturing $\mathcal{ALC}$ axioms. For each atomic concept and role in $\mathbf{KB}_M$ we introduce an entity and a relationship, respectively. To simulate the universal concept, $\top$, we introduce a snapshot entity, `Top`, that generalizes all the entities in Σ_M. Additionally, the various axioms in $\mathbf{KB}_M$ are encoded in $\mathcal{ER}_{VT}$ 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 ISA D.

3. For axioms of the form $C \sqsubseteq \neg D$ we construct the hierarchy in Figure 4(a).

4. For axioms of the form $C \sqsubseteq D_1 \sqcup \ldots \sqcup D_n$ 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 role[3]. Figure 4(c) 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 $C \sqsubseteq \Box^+ D$ ($C \sqsubseteq \Diamond^+ D$) we use a persistency (respectively, dynamic extension) constraint: C PER D (respectively, C DEX D).

7. Axioms of the form $\mathtt{next}(C, D)$ are mapped by using the dynamic extension constraint to capture that $C \sqsubseteq \Diamond^+ D$. To capture that $C \sqsubseteq \neg\Diamond^+\Diamond^+ D$ we rewrite it as $C \sqsubseteq \Box^+\Box^+\neg D$, which, in turn, is encoded by the following axioms: $C \sqsubseteq \Box^+ C_1$; $C_1 \sqsubseteq \Box^+ C_2$; $C_2 \sqsubseteq \neg D$. Figure 4(d) shows the diagram that maps `next` axioms.

[3] Considering R as a functional role does not change the $\mathcal{ALC}_\mathsf{F}$ undecidability proof.

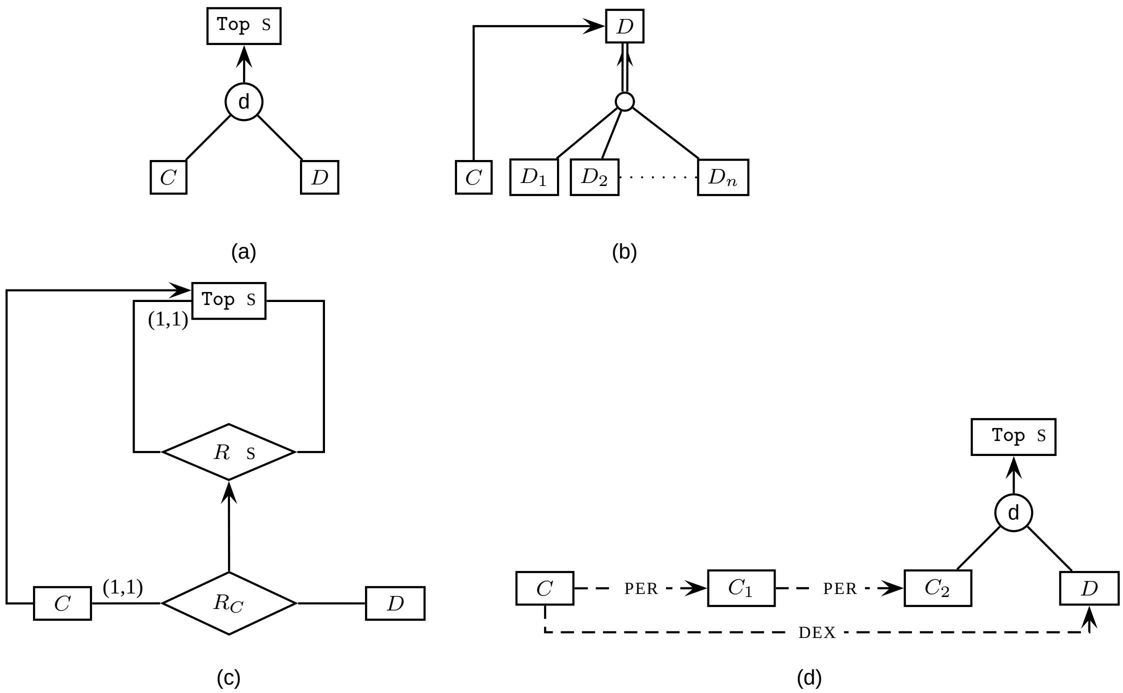

Figure 4: Encoding axioms: (a) $C \sqsubseteq \neg D$; (b) $C \sqsubseteq D_1 \sqcup \ldots \sqcup D_n$; (c) $C \sqsubseteq \forall R.D$ and $\top \sqsubseteq \exists R.\top$; (d) $\texttt{next}(C, D)$.

The above reductions are enough to capture all axioms in $\mathbf{KB}_M$. Indeed, axioms (11–13) have the form: $C \sqsubseteq \neg C_1 \sqcup \forall R.C_2$, while axioms (16) have the form: $C_a \sqsubseteq C_l \sqcup C_s \sqcup C_r \sqcup \forall R.C_a$. We are now able to prove the main result of this paper.

Theorem 5.2. *Reasoning in $\mathcal{ER}_{VT}$ using persistency and dynamic constructs is undecidable.*

Proof. Proving that the above reduction from $\mathbf{KB}_M$ to Σ_M is true can be easily done by checking the semantic equivalence between each $\mathcal{ALC}_{\mathsf{F}}$ axiom and its encoding (for a similar proof see [6]). Then, the concept C_0 is satisfiable w.r.t. $\mathbf{KB}_M$ iff the entity C_0 is satisfiable w.r.t. Σ_M. Thus, because of Proposition 5.1, the halting problem can be reduced to reasoning in $\mathcal{ER}_{VT}$. $\square$

Acknowledgments

I would like to thank Diego CAlvanese, Enrico Franconi, Sergio Tessaris and Frank Wolter together with the anonymous referees for enlightening comments on earlier drafts of the paper.

References

[1] A. Artale. Reasoning on temporal conceptual schemas with dynamic constraints. In *Proc. of the 11th Int. Symposium on Temporal Representation and Reasoning, TIME'04*. IEEE Computer Society, July 2004.

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

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

[4] 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*. LNCS, Springer, 2003.

[5] Alessandro Artale, Enrico Franconi, Frank Wolter, and Michael Zakharyaschev. A temporal description logic for reasoning about conceptual schemas and queries. In S. Flesca, S. Greco, N. Leone, and G. Ianni, editors, *Proc. of the 8th Joint European Conference on Logics in Artificial Intelligence (JELIA-02)*, volume 2424 of *LNAI*. Springer, 2002.

[6] Daniela Berardi, Andrea Calì, Diego Calvanese, and Giuseppe De Giacomo. Reasoning on UML class diagrams. Technical Report 11-03, 2003.

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

[8] J. Chomicki and D. Toman. Temporal logic in information systems. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*. Kluwer, 1998.

[9] D. Gabbay, A.Kurucz, F. Wolter, and M. Zakharyaschev. *Many-dimensional modal logics: theory and applications*. Studies in Logic. Elsevier, 2003.

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

[11] 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*. Springer-Verlag, 1998.

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

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

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

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

[16] 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.