Reasoning on Temporal Conceptual Schemas with Dynamic Constraints

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

Abstract

This paper formally clarifies the relevant reasoning problems for temporal EER diagrams. We distinguish between the following reasoning services: (a) Entity, relationship and schema satisfiability; (b) Liveness and global satisfiability for both entities and relationships; (c) Subsumption for either entities or relationships; (d) Logical implication between schemas. We then show that reasoning on temporal models is an undecidable problem as soon as the schema language is able to distinguish between temporal and atemporal constructs, and it has the ability to represent dynamic constraints between entities.

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, 3], and to check the possibility to perform *reasoning* on top of temporal schemas [4]. $\mathcal{ER}_{VT}$ is equipped with both a linear and a graphical syntax along with a model-theoretic semantics. It 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 [3] 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 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 [4] showed that $\mathcal{ER}_{VT}$ diagrams can be embedded into the temporal description logic $\mathcal{DLR}_{\mathcal{US}}$—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. 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. Note that, 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] by showing that checking concept satisfiability w.r.t. an $\mathcal{ALC}_\mathsf{F}$ KB made by just simple global axioms is an undecidable problem.

The paper in organized as follows. The temporal description logic $\mathcal{ALC}_\mathsf{F}$ 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. Section 6 makes final conclusions and mention an interesting open problem.

2. The Temporal Description Logic

In this Section we introduce the $\mathcal{ALC}_\mathsf{F}$ description logic [16, 1, 9] as a the tense-logical extension of $\mathcal{ALC}$. Basic types of $\mathcal{ALC}_\mathsf{F}$ 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 represented by means of roles, which are interpreted as binary

* The author has been partially supported by the EU projects Sewasie, KnowledgeWeb, and Interop.

1 EER is the standard entity-relationship data model, enriched with ISA links, generalized hierarchies with disjoint and covering constraints, and full cardinality constraints [8].

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

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

relations over the domain of individuals. According to the syntax rules of Figure 1, $\mathcal{ALC}_{\mathsf{F}}$ *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 $\Box^+$ (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 (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 Figure 1—note that the operator $\Box^+$ is the dual of $\Diamond^+$, i.e. $\Box^+ C \equiv \neg\Diamond^+\neg C$.

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 $t \in \mathcal{T}$. 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. $\Sigma \not\models C \sqsubseteq \bot$.

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, i.e. temporary), 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 [3]).

$\mathcal{ER}_{VT}$ is equipped with both a linear and a graphical syntax along with a model-theoretic semantics as a temporal extension of the EER semantics [6]. 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 $T[x_i] = y_i$. In the following definition we refer to Figure 2 to show the visual syntax associated to the various $\mathcal{ER}_{VT}$ constructs.

Definition 3.1 ($\mathcal{ER}_{VT}$ Syntax). *An $\mathcal{ER}_{VT}$ schema is a tuple:*
$\Sigma = (\mathcal{L}, \text{REL}, \text{ATT}, \text{CARD}, \text{ISA}, \text{DISJ}, \text{COVER}, \text{S}, \text{T}, \text{KEY}, \text{DEX}, \text{PER})$, *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). We will call the tuple $(\mathcal{E}, \mathcal{A}, \mathcal{R}, \mathcal{U}, \mathcal{D})$ the* signature *of the schema Σ. $\mathcal{E}$ 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* implicitly 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. If $\text{REL}(R) = \langle U_1 : E_1, \ldots, U_k : E_k \rangle$, then $\text{CARD}(E, R, U)$ is defined only if $U = U_i$ and $E = E_i$, for some $i \in \{1, \ldots, k\}$. We denote with $\text{CMIN}(E, R, U)$ and $\text{CMAX}(E, R, U)$ the first and second component of* CARD. *If not stated otherwise,* CMIN *is assumed to be zero, and* CMAX *is assumed to be ∞. 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 arity.* ISA *is visualized with a directed arrow, e.g.* `Manager` ISA `Employee` *in Figure 2.*

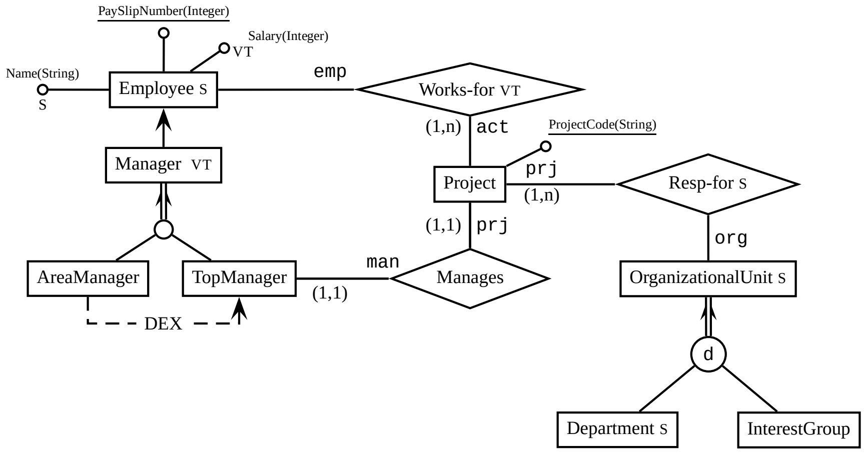


Figure 2. An $\mathcal{ER}_{VT}$ diagram

$\textsc{disj}, \textsc{cover}$ *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. Furthermore, if* $\langle E, A\rangle \in \textsc{s}, \textsc{t}$, *then* A *is between the attributes in* $\textsc{att}(E)$ *(see* S, T *marked attributes in Figure 2).*

KEY *is a function that maps entity symbols in* $\mathcal{E}$ *to their key attributes,* $\textsc{key}(E) = A$. *Furthermore, if* $\textsc{key}(E) = A$, *then* A *is between the attributes in* $\textsc{att}(E)$. *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*[2]. 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*[3] representation of abstract temporal databases and temporal conceptual models [7]. 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 3.2 ($\mathcal{ER}_{VT}$ Semantics). *Let* Σ *be an* $\mathcal{ER}_{VT}$ *schema, and* $BD = \bigcup_{D_i \in \mathcal{D}} BD_i$ *be a set of* basic domains *such that* $BD_i \cap BD_j = \emptyset$ *for* $i \neq j$. *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 *such that* $\Delta^{\mathcal{B}}_{D_i} \subseteq BD_i$*—we call* $\Delta^{\mathcal{B}}_{D_i}$ *the* active domain*;* $\cdot^{\mathcal{B}(t)}$ *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 of the integrity constraints expressed in the schema:*

- *For each* $E_1, E_2 \in \mathcal{E}$, *if* E_1 ISA E_2, *then,* $E_1^{\mathcal{B}(t)} \subseteq E_2^{\mathcal{B}(t)}$.
- *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* $E \in \mathcal{E}$, *if* $\textsc{att}(E) = \langle A_1 : D_1, \ldots, A_h : D_h\rangle$, *then,* $e \in E^{\mathcal{B}(t)} \rightarrow (\forall i \in \{1, \ldots, h\}, \exists! a_i.\ \langle e, a_i\rangle \in A_i^{\mathcal{B}(t)} \land \forall a_i.\langle e, a_i\rangle \in A_i^{\mathcal{B}(t)} \rightarrow a_i \in \Delta^{\mathcal{B}}_{D_i})$.
- *For each* $R \in \mathcal{R}$, *if* $\textsc{rel}(R) = \langle U_1 : E_1, \ldots, U_k : E_k\rangle$, *then,* $r \in R^{\mathcal{B}(t)} \rightarrow (r = \langle U_1 : e_1, \ldots, U_k : e_k\rangle \land \forall i \in \{1, \ldots, k\}.e_i \in E_i^{\mathcal{B}(t)})$. *In the following, we adopt the convention:* $\langle U_1 : e_1, \ldots, U_k : e_k\rangle \equiv \langle e_1, \ldots, e_k\rangle$, *and* $r[U_i] \equiv r[i]$ *to denote the* U_i/i*-component of* r.
- *For each cardinality constraint* $\textsc{card}(E, R, U)$, *then,* $e \in E^{\mathcal{B}(t)} \rightarrow \textsc{cmin}(E, R, U) \leq \#\{r \in R^{\mathcal{B}(t)} \mid r[U] = e\} \leq \textsc{cmax}(E, R, U)$.
- *For each snapshot entity* $E \in \mathcal{E}^S$, *then,* $e \in E^{\mathcal{B}(t)} \rightarrow \forall t' \in \mathcal{T}.e \in E^{\mathcal{B}(t')}$.
- *For each temporary entity* $E \in \mathcal{E}^T$, *then,* $e \in E^{\mathcal{B}(t)} \rightarrow \exists t' \neq t.e \notin E^{\mathcal{B}(t')}$.
- *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 each entity* $E \in \mathcal{E}$ *with a snapshot attribute* A_i, *i.e.* $\langle E, A_i\rangle \in \textsc{s}$, *then,* $(e \in E^{\mathcal{B}(t)} \land \langle e, a_i\rangle \in A_i^{\mathcal{B}(t)}) \rightarrow \forall t' \in \mathcal{T}.\langle e, a_i\rangle \in A_i^{\mathcal{B}(t')}$.
- *For each entity* $E \in \mathcal{E}$ *with a temporary attribute* A_i, *i.e.* $\langle E, A_i\rangle \in \textsc{t}$, *then,* $(e \in E^{\mathcal{B}(t)} \land \langle e, a_i\rangle \in A_i^{\mathcal{B}(t)}) \rightarrow \exists t' \neq t.\langle e, a_i\rangle \notin A_i^{\mathcal{B}(t')}$.
- *For* $E, E_1, \ldots, E_n \in \mathcal{E}$,
  - *If* $\{E_1, \ldots, E_n\}$ DISJ E, *then,* $\forall i \in \{1, \ldots, n\}.E_i$ ISA $E \land$ $\forall j \in \{1, \ldots, n\}, j \neq i.E_i^{\mathcal{B}(t)} \cap E_j^{\mathcal{B}(t)} = \emptyset$.

2 For ISA relations, we use the notation E_1 ISA E_2 instead of $\langle E_1, E_2\rangle \in$ ISA. Similarly for DISJ,COVER,DEX,PER.

3 The snapshot model represents the same class of temporal databases as the *timestamp* model [12, 13] defined by adding temporal attributes to a relation [7].

– If $\{E_1,\ldots,E_n\}$ COVER E, then, $\forall i \in \{1,\ldots,n\}.E_i$ ISA $E \wedge E^{\mathcal{B}(t)} = \bigcup_{i=1}^{n} E_i^{\mathcal{B}(t)}$.

- *For each $E \in \mathcal{E}, A \in \mathcal{A}$ such that* KEY$(E) = A$, *then,* $\langle E, A_i\rangle \in$ 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, then,* $e \in E_1^{\mathcal{B}(t)} \rightarrow \exists t_1 > t.e \in E_2^{\mathcal{B}(t_1)}$;
  – *If E_1* PER *E_2, then,* $e \in E_1^{\mathcal{B}(t)} \rightarrow \forall t' > t.e \in E_2^{\mathcal{B}(t')}$.

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. We start with the formal definition of the relevant reasoning services in a temporal schema as presented in [3]. Based on this formal characterization we can prove the first results of this paper concerning reasoning in $\mathcal{ER}_{VT}$: a) Subsumption and satisfiability reasoning services relative to entities are mutually reducible to each other; b) Satisfiability problems relative to relationships are mutually reducible; c) Satisfiability of relationships reduces to satisfiability of entities and viceversa; d) Logical implication is the more general service.

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'.*

We now prove that reasoning services (1-5) relative to entities and knowledge bases are mutually reducible to each other.

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

Proof *Proving the mutual reducibility between satisfiability and subsumption in $\mathcal{ER}_{VT}$ can be done similarly to [5]. Then, in the following we prove that given an $\mathcal{ER}_{VT}$ schema Σ:*

1. *Entity satisfiability reduces to schema satisfiability;*
2. *Schema satisfiability reduces to entity liveness satisfiability;*
3. *Entity liveness satisfiability reduces to entity global satisfiability;*
4. *Entity global satisfiability reduces to entity satisfiability.*

(1) *We prove that given an entity $E_0 \in \mathcal{E}$, then, 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.*

"$\Leftarrow$" Let Σ' be satisfiable, then, Σ' has a model $\mathcal{B}$ (which is a model for Σ, too) such that $\exists t \in \mathcal{T}.\exists e \in \Delta^{\mathcal{B}}.e \in \top^{\mathcal{B}(t)}$ (by definition of schema satisfiability and by construction of $\top$ as superclass of all entities in Σ'). Because $\top$ is a snapshot entity, then, $\forall t \in \mathcal{T}.e \in \top^{\mathcal{B}(t)}$. Since E_1, E_2 form a disjoint covering of $\top$, and E_1, E_2 are both temporary, then, $\exists t' \in \mathcal{T}.e \in E_1^{\mathcal{B}(t')}$. Finally, because E_1 totally participates in R, then, $\exists e_0 \in \Delta^{\mathcal{B}}.(\langle e, e_0\rangle \in R^{\mathcal{B}(t')} \wedge e_0 \in E_0^{\mathcal{B}(t')})$. Then, E_0 is satisfiable w.r.t. Σ.

"$\Rightarrow$" Let E_0 be satisfiable w.r.t. Σ, then, Σ has a model $\mathcal{B}$ such that $\exists t_0 \in \mathcal{T}.\exists e_0 \in \Delta^{\mathcal{B}}.e_0 \in E_0^{\mathcal{B}(t_0)}$. We now construct a model $\mathcal{B}'$ for Σ'. Let $\mathcal{B}$ and $\mathcal{B}'$ coincide on all constructs in Σ, and additionally, for all $t \in \mathcal{T}$:

- $\top^{\mathcal{B}'(t)} = \bigcup_{v \in \mathcal{T}} \bigcup_{E \in \mathcal{E}} E^{\mathcal{B}(v)}$
- $E_1^{\mathcal{B}'(t)} = \begin{cases} \top^{\mathcal{B}'(t)} & \text{if } t = t_0 \\ \emptyset & \text{otherwise} \end{cases}$
- $E_2^{\mathcal{B}'(t)} = \top^{\mathcal{B}'(t)} \setminus E_1^{\mathcal{B}'(t)}$
- $R^{\mathcal{B}'(t)} = \{\langle e, e_0\rangle \mid e \in E_1^{\mathcal{B}'(t)}\}$

It is easy to check that $\mathcal{B}'$ is a model for Σ', then, Σ' is satisfiable.

(2) *We prove that a given 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, we prove that Σ is satisfiable iff $\top_1$ is liveness satisfiable w.r.t. Σ'.*

"$\Leftarrow$" Let $\top_1$ be liveness satisfiable w.r.t. Σ'. Then, Σ' has a model, $\mathcal{B}$, such that $\forall t \in \mathcal{T}.\exists t' > t.\exists o \in \Delta^{\mathcal{B}}.o \in \top_1^{\mathcal{B}(t')}$. Since $\top_1$ is a snapshot entity, then, $o \in \top_1^{\mathcal{B}(t)}$, for all $t \in \mathcal{T}$. Because E_1, E_2 are a disjoint covering of $\top_1$ and they are both temporary, then, $\exists \bar{t} \in \mathcal{T}.o \in E_1^{\mathcal{B}(\bar{t})}$. Because E_1 totally participates in R, then, $\exists e \in \Delta^{\mathcal{B}}.(\langle o, e\rangle \in R^{\mathcal{B}(\bar{t})} \wedge e \in \top_2^{\mathcal{B}(\bar{t})})$. Then, $\top_2$ is a satisfiable entity and, because of the covering constraint, Σ is satisfiable.

"$\Rightarrow$" Let Σ be a satisfiable schema and $\mathcal{B}$ a model for Σ. We now show how to build a model, $\mathcal{B}'$, for Σ' such that $\top_1$ is liveness satisfiable. $\mathcal{B}'$ agrees with $\mathcal{B}$ on all constructs in Σ, and additionally, for all $t \in \mathcal{T}$:

- $\top_1^{\mathcal{B}'(t)} = \bigcup_{v \in \mathcal{T}} \top_2^{\mathcal{B}(v)}$
  Note that, because by assumption Σ is satisfiable, then, $\top_2$ is satisfiable while $\top_1$ contains always at least one element (i.e., it is globally, and then liveness, satisfiable).

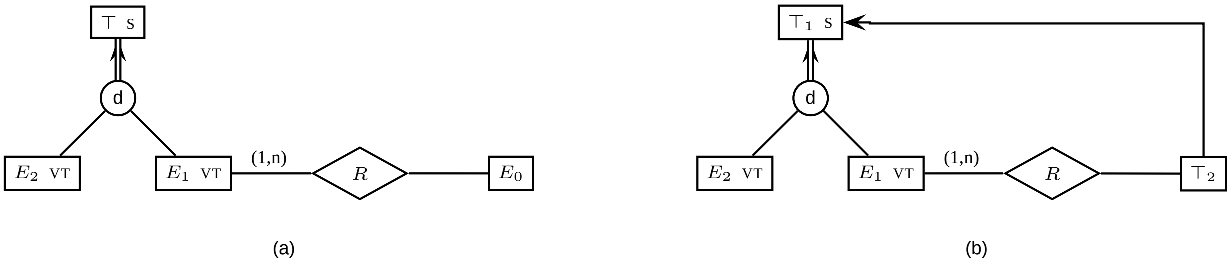


Figure 3. Reductions: (a) From Entity Sat to Schema Sat; (b) From Schema Sat to Entity Liveness Sat.

- *Let $t_0 \in \mathcal{T}$ an arbitrary time such that $\exists e_0 \in \Delta^{\mathcal{B}}.e_0 \in \top_2^{\mathcal{B}(t_0)}$, then:*
  $E_1^{\mathcal{B}'(t)} = \begin{cases} \top_1^{\mathcal{B}'(t)} & \text{if } t = t_0 \\ \emptyset & \text{otherwise} \end{cases}$
- $E_2^{\mathcal{B}'(t)} = \top_1^{\mathcal{B}'(t)} \setminus E_1^{\mathcal{B}'(t)}$
- $R^{\mathcal{B}'(t)} = \{\langle e, e_0\rangle \mid e \in E_1^{\mathcal{B}'(t)}\}$

Then, $\mathcal{B}'$ is a model of Σ' such that $\top_1$ is liveness satisfiable.

(3) *We prove that given an entity $E_0 \in \mathcal{E}$, then, 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 showed in figure 4(a). We prove that E_0 is liveness satisfiable w.r.t. Σ iff E_1 is globally satisfiable w.r.t. Σ'.*

"$\Leftarrow$" Let E_1 be globally satisfiable w.r.t. Σ'. Then, Σ' has a model, $\mathcal{B}$, such that $\forall t \in \mathcal{T}.\exists o \in \Delta^{\mathcal{B}}.o \in E_1^{\mathcal{B}(t)}$. Then, given the dynamic extension constraint between E_1 and E_0, E_0 is liveness satisfiable.

"$\Rightarrow$" Let E_0 be liveness satisfiable w.r.t. Σ. Then, Σ has a model, $\mathcal{B}$, such that $\forall t \in \mathcal{T}.\exists t' > t.\exists e \in \Delta^{\mathcal{B}}.e \in E_0^{\mathcal{B}(t')}$. We now extend $\mathcal{B}$ to E_1, such that for all $t \in \mathcal{T}$:

- $E_1^{\mathcal{B}(t)} = \{e \in \Delta^{\mathcal{B}} \mid \exists t' > t.e \in E_0^{\mathcal{B}(t')}\}$

Then, $\mathcal{B}$ is a model of Σ' such that E_1 is globally satisfiable.

(4) *We prove that given an entity $E_0 \in \mathcal{E}$, then, E_0 is globally satisfiable w.r.t. Σ iff an entity is satisfiable w.r.t. a new schema Σ'. Σ' is obtained by adding to Σ the schema in figure 4(b), where E_1 is new snapshot entity and R is a new binary relationship.*

"$\Leftarrow$" Let E_1 be satisfiable w.r.t. Σ', then, Σ' has a model $\mathcal{B}$ such that $\exists o \in \Delta^{\mathcal{B}}.o \in E_1^{\mathcal{B}(t)}$, for all $t \in \mathcal{T}$ (by construction of E_1 as a snapshot entity). Since E_1 totally participates in R, then, $\exists e \in \Delta^{\mathcal{B}}.\langle o, e\rangle \in R^{\mathcal{B}(t)} \wedge e \in E_0^{\mathcal{B}(t)}$. Since this must be true at all time, then, E_0 is globally satisfiable w.r.t. Σ.

"$\Rightarrow$" Let E_0 be globally satisfiable w.r.t. Σ. Then, Σ has a model, $\mathcal{B}$, such that $\forall t \in \mathcal{T}.\exists e \in \Delta^{\mathcal{B}}.e \in E_0^{\mathcal{B}(t)}$. We now construct a model, $\mathcal{B}'$, for Σ' such that E_1 is satisfiable. $\mathcal{B}'$ agrees with $\mathcal{B}$ on all constructs in Σ, and additionally, for all $t \in \mathcal{T}$:

- $E_1^{\mathcal{B}'(t)} = \bigcup_{v \in \mathcal{T}} E_0^{\mathcal{B}(v)}$
- $R^{\mathcal{B}'(t)} = \{\langle o, e\rangle \mid o \in E_1^{\mathcal{B}'(t)} \wedge e \in E_0^{\mathcal{B}(t)}\}$

Then, $\mathcal{B}'$ is a model of Σ' such that E_1 is satisfiable.

❑

We are now able to prove that satisfiability problems for relationships are reducible to the same problems for entities and viceversa.

Proposition 4.3. *There is a mutual reducibility between the reasoning services (1-4) relative to both relationships and entities in $\mathcal{ER}_{VT}$.*

Proof *We only prove that satisfiability of relationships can be reduced to satisfiability of entities and viceversa. The other mutual reductions easily follow from analogous results proved in Proposition 4.2.*

"R SAT reduces to E SAT." 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 if and only if A_R is satisfiable.

"E SAT reduces to R SAT." We can verify whether an entity E is satisfiable in Σ by adding a new relationship, say R_E such that: (a) R_E is a binary relationship with both arguments restricted to E; (b) E totally participates in R_E. Easily follows that E is satisfiable if and only if R_E is satisfiable.

❑

Finally, we show that all the reasoning problems can be reduced to a logical implication problem. Logical implication accounts for checking properties of a schema whenever they can be expressed in the $\mathcal{ER}_{VT}$ schema language. In particular, 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 $\Sigma' = \{R_1 \text{ ISA } R_2\}$. This shows that logical implication is the most general reasoning service.

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 product 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 $\mathcal{ALC}_{\mathsf{F}}$ KB;
3. Reduction of concept satisfiability w.r.t. an $\mathcal{ALC}_{\mathsf{F}}$ KB to entity satisfiability w.r.t. an $\mathcal{ER}_{VT}$ schema.

(a) (b)

Figure 4. Reductions: (a) From Entity Liveness Sat to Entity Global Sat; (b) From Entity Global Sat to Entity Sat.

The second step has been chosen as an intermediate step to better understand the halting problem reduction by using the concise $\mathcal{ALC}_{\mathsf{F}}$ linear syntax. Then, the final step will show how $\mathcal{ER}_{VT}$ is able to capture the $\mathcal{ALC}_{\mathsf{F}}$ axioms used in the reduction.

Halting problem

We show here a formal representation of the halting problem for Turing machines as presented in [9]. 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*, $\rho : (S - \{s_1\}) \times A \rightarrow S \times (A \cup \{\mathtt{L}, \mathtt{R}\})$. A *Configuration* of $\mathbf{M}$ is an infinite sequence: $\langle \pounds, a_1, \ldots, a_{i-1}, \langle s_i, a_i\rangle, \ldots, a_n, b, \ldots\rangle$, where, $\pounds \notin A$ 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 $(A \cup \{\pounds\}) \times ((S - \{s_1\}) \times A) \times A$, such that:

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

A sequence $\langle c_0, c_1, \ldots, c_k, c_{k+1}, \ldots\rangle$ of configurations of $\mathbf{M}$ is said 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 living the rest unaltered. We say that $\mathbf{M}$ *halts*, starting with the empty tape—i.e. with starting configuration: $\langle \pounds\langle s_0, b\rangle, b, \ldots, b, \ldots\rangle$—if there is a finite computation, $\langle c_0, c_1, \ldots, c_k\rangle$, such that the state of c_k is s_1 (the final state).

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]. The following theorem proves that checking concept satisfiability w.r.t. an $\mathcal{ALC}_{\mathsf{F}}$ KB made by just simple global axioms is an undecidable problem.

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

Proof *Given a Turing machine, $\mathbf{M} = \langle A, S, \rho\rangle$, 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 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 $\mathtt{next}(C, D)$ as the following axiom: $C \sqsubseteq \Diamond^+ D \sqcap \neg\Diamond^+\Diamond^+ D$. 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 $C, D_1, \ldots, D_n$ concepts, $\mathtt{discover}(C, \{D_1, \ldots, D_n\})$ is defined as the conjunction of the following axioms:*

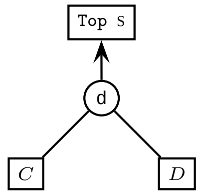
$C \sqsubseteq D_1 \sqcup \ldots \sqcup D_n$
$D_1 \sqsubseteq C \sqcap \neg D_2 \sqcap \ldots \sqcap \neg D_n$
$\ldots$
$D_{n-1} \sqsubseteq C \sqcap \neg D_n$

i.e., there is a disjoint covering between C and $D_1 \ldots D_n$. Let $A' = A \cup \{\pounds\} \cup (S \times A)$. 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:

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

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), $x_0 \in C_\pounds^{\mathcal{I}(t_0)}$, and $\exists \bar{t} > t_0 . C_{\langle s_0, b\rangle}^{\mathcal{I}(\bar{t})}$. 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), $C_\pounds$ 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, $\bar{t} = t_0 + 1$, $x_0 \in C_\pounds^{\mathcal{I}(t_0)}, x_0 \in C_{\langle s_0, b\rangle}^{\mathcal{I}(t_0+1)}$, 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), $x_0 \in C_l^{\mathcal{I}(t_0)}, x_0 \in C_r^{\mathcal{I}(t_0+2)}$. Because by axiom (2), for all



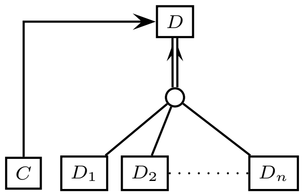


Figure 5. Encoding axioms: (a) $C \sqsubseteq \neg D$; (b) $C \sqsubseteq D_1 \sqcup \ldots \sqcup D_n$.

$t \in \mathcal{T}$ there is at most one $x \in A'$ such that $x_0 \in C_x^{\mathcal{I}(t)}$, then, the sequence $\langle\langle x_0, t_0\rangle, \langle x_0, t_0+1\rangle, \ldots\rangle$ represents 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 $\langle x_0, x_1, x_2, \ldots\rangle$ be a chain of R-successors satisfying axiom (2). Since $x_0 \in C_{\pounds}^{\mathcal{I}(t_0)}$, then, by axioms (13) and (16), and the definition of the instruction function, δ, $x_i \in C_{\pounds}^{\mathcal{I}(t_0)}$, for all i. Then, given the axioms (12–16), the chain of R-successor, $\langle x_0, x_1, x_2, \ldots\rangle$, 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 $\langle c_0, \ldots, c_k, \ldots\rangle$ its infinite 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 $\mathbf{KB}_M$ such that C_0 is satisfiable. In particular, we fix $\mathcal{T} = \langle \mathbb{N}, <\rangle$[4], $\Delta^{\mathcal{I}} = \mathbb{N}$, $R^{\mathcal{I}} = \mathtt{suc}_{\mathbb{N}}$ (the successor function over $\mathbb{N}$), $C_0^{\mathcal{I}(0)} = \{0\}$, and $C_0^{\mathcal{I}(j)} = \emptyset$, for all $j > 0$. Furthermore, $\forall j \in \mathbb{N}$:

- $C_x^{\mathcal{I}(j)} = \{i \in \mathbb{N} \mid$ *the jth cell of c_i contains x*$\}$, *for all* $x \in A'$
- $C_s^{\mathcal{I}(j)} = \{i \in \mathbb{N} \mid$ *the jth cell of c_i is the active one*$\}$
- $C_l^{\mathcal{I}(j)} = C_s^{\mathcal{I}(j+1)}$
- $C_r^{\mathcal{I}(j)} = C_s^{\mathcal{I}(j-1)}$
- $C^{\mathcal{I}(j)} = \bigcup_{x \in A'} C_x^{\mathcal{I}(j)}$
- $D_1^{\mathcal{I}(j)} = C_{\pounds}^{\mathcal{I}(j-1)}$
- $D_2^{\mathcal{I}(j)} = D_1^{\mathcal{I}(j-1)}$
- $D_3^{\mathcal{I}(j)} = C_r^{\mathcal{I}(j-1)}$
- $S1^{\mathcal{I}(j)} = \bigcup_{a \in A} C_{\langle s_1, a\rangle}^{\mathcal{I}(j)}$.

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

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

We now show how to capture the knowledge base $\mathbf{KB}_M$ with an $\mathcal{ER}_{VT}$ schema, Σ_M. The mapping is based on a similar reduction presented in [5] 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:

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

1. Axioms involving `discover` are mapped using disjoint and covering hierarchies in $\mathcal{ER}_{VT}$.
2. Axioms of the form $C \sqsubseteq D$, with C, D atomic concepts are encoded as C ISA D.
3. For each axiom of the form $C \sqsubseteq \neg D$ we construct the hierarchy in Figure 5(a).
4. For each axiom of the form $C \sqsubseteq D_1 \sqcup \ldots \sqcup D_n$ we introduce a new entity, D, and then we construct the hierarchy in Figure 5(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 subrelationship, R_C, and considering R as a functional role[5]. Figure 6(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 $C \sqsubseteq \Box^+ D$ ($C \sqsubseteq \Diamond^+ D$) we use a persistency (dynamic extension) constraint: C PER D (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 set of axioms:
   $$C \sqsubseteq \Box^+ C_1$$
   $$C_1 \sqsubseteq \Box^+ C_2$$
   $$C_2 \sqsubseteq \neg D$$
   Figure 6(b) shows the portion of the $\mathcal{ER}_{VT}$ diagram that maps `next` axioms.

The above reductions are enough to capture all axioms in $\mathbf{KB}_M$. Indeed, axioms (13–15) have the form: $C \sqsubseteq \neg C_1 \sqcup \forall R.C_2$. They can be split by introducing new concepts $\overline{C}_1, \overline{C}_2$ as follows:

$$C \sqsubseteq \overline{C}_1 \sqcup \overline{C}_2$$
$$\overline{C}_1 \sqsubseteq \forall R.C_1$$
$$\overline{C}_2 \sqsubseteq \neg C_2$$

We proceed in a similar way to encode axioms (16) which have the form: $C_a \sqsubseteq C_l \sqcup C_s \sqcup C_r \sqcup \forall R.C_a$, and the axiom (12). 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 [5]). 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*

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

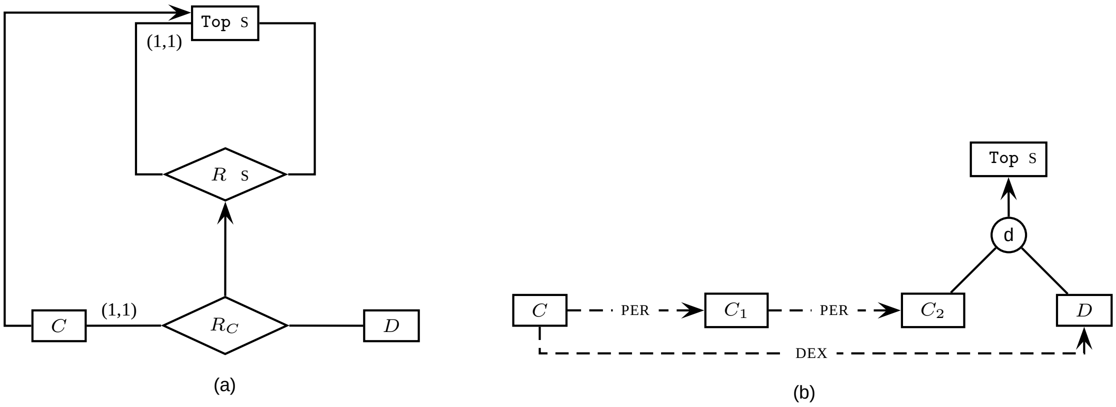


Figure 6. Encoding axioms: (a) $C \sqsubseteq \forall R.D$ and $\top \sqsubseteq \exists R.\top$; (b) $\mathtt{next}(C, D)$.

of Proposition 5.1, the halting problem can be reduced to reasoning in $\mathcal{ER}_{VT}$. ❑

6. Conclusions

We formally discussed the relevant reasoning problems for temporal conceptual models. We distinguished between six different reasoning services: (a) Entity, relationship and schema satisfiability; (b) Liveness and global satisfiability for both entities and relationships; (c) Subsumption for either entities or relationships; (d) Logical implication between schemas. While the problems (a-c) have been shown to be reducible to each other, checking whether a schema logically implies another schema has been shown to be the more general reasoning service.

We then investigated the complexity of reasoning on temporal models and we found that such problem is undecidable as soon as the schema 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 entities.

We finally mention an interesting open problem which will be matter of a future work. Does reasoning on $\mathcal{ER}_{VT}$ become decidable if we drop dynamic constraints? Without dynamic constraints it is possible to encode $\mathcal{ER}_{VT}$ using a combination between the description logic $\mathcal{ALCQI}$ and the epistemic modal logic S5. Decidability results have been proved for the logic $\mathcal{ALC}_{S5}$ [9]. But, it is still an open problem whether this result holds for the more complex logic $\mathcal{ALCQI}_{S5}$.

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

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

[3] A. Artale, E. Franconi, and F. Mandreoli. Description logics for modelling dynamic information. In J. Chomicki, R. van der Meyden, and G. Saake, editors, *Logics for Emerging Applications of Databases*. Lecture Notes in Computer Science, Springer-Verlag, 2003.

[4] A. Artale, E. Franconi, F. Wolter, and M. Zakharyaschev. 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.

[5] D. Berardi, A. Calì, D. Calvanese, and G. De Giacomo. Reasoning on UML class diagrams. Technical Report 11-03, 2003.

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

[7] 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.

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

[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. 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. J. 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.

[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.