A Description Logic of Change*

Alessandro Artale University of Bolzano, Italy artale@inf.unibz.it
Carsten Lutz University of Dresden, Germany clu@tcs.inf.tu-dresden.de
David Toman University of Waterloo, Canada david@uwaterloo.ca

Abstract

We combine the modal logic S5 with the description logic (DL) $\mathcal{ALCQI}$. The resulting multi-dimensional DL $\mathsf{S5}_{\mathcal{ALCQI}}$ supports reasoning about change by allowing to express that concepts and roles change over time. It cannot, however, discriminate between changes in the past and in the future. Our main technical result is that satisfiability of $\mathsf{S5}_{\mathcal{ALCQI}}$ concepts with respect to general TBoxes (including GCIs) is decidable and 2-ExpTime-complete. In contrast, reasoning in temporal DLs that *are* able to discriminate between past and future is inherently undecidable. We argue that our logic is sufficient for reasoning about temporal conceptual models with time-stamping constraints.

1 Introduction

An important application of Temporal Description Logics (TDLs) is the representation of and reasoning about temporal conceptual models [Artale, 2004; Artale *et al.*, 2003; 2002]. Knowledge captured by such models is translated into a TDL TBox and reasoning algorithms for TDL are then used to detect inconsistencies and implicit IS-A relations in the temporal model [Artale and Franconi, 1999; Artale *et al.*, 2002; Calvanese *et al.*, 1998]. A serious obstacle for putting this general idea to work is posed by the fact that for many natural temporal conceptual formalisms and their associated TDLs, reasoning turns out to be undecidable.

The most prominent witnesses of this problem are the various temporal entity-relationship (TER) models used to design temporal databases [Chomicki and Toman, 2005]. TERs are classical ER data models extended with two additional classes of constraints that model the temporal evolution of data in an application domain [Spaccapietra *et al.*, 1998]. First, *timestamping constraints* are used to distinguish temporal and atemporal components of a TER model. Timestamping is usually implemented by marking entities (i.e., classes), relationships and attributes as *snapshot*, *temporary*, or *unrestricted*. The idea behind such a classification is to express that object membership in entities, relationships, and attribute values cannot or must change in time; this is achieved by snapshot and temporary marks in the diagram, respectively. Second, *evolution constraints* govern object migration between entities and can state, for example, that every instance of the entity Child will eventually become an instance of the entity Adult.

TER models with both timestamping and evolution constraints can be translated into the TDL $\mathsf{DLR}_{\mathcal{US}}$ [Artale *et al.*, 2002]. Unfortunately, reasoning in this logic is undecidable. Moreover, the computational problems are not due to the translation to TDLs: even direct reasoning in the generally less powerful TER models is undecidable [Artale, 2004]. There are two principal ways around this problem. The first approach restricts the application of timestamping: it allows arbitrary timestamping of entities, but gives up timestamping of relationships and attributes (i.e., all relationships and attributes are unrestricted). This re-establishes decidability of TER models with restricted timestamping and evolution constraints [Artale *et al.*, 2002]. The second approach to regaining decidability allows for full use of timestamping, but prohibits the use of evolution constraints.

This second alternative is pursued in the current paper. We devise a multi-dimensional description logic $\mathsf{S5}_{\mathcal{ALCQI}}$ that is obtained by combining the modal logic S5 with the standard DL $\mathcal{ALCQI}$. The S5 modality can be applied to both concepts and roles; axioms in the TBox are, however, interpreted globally. This logic can be viewed as a *description logic of change*: it can express that concept and role memberships change in time, but does not permit discriminating between changes in the past or future. We show that TER models with full timestamping (i.e., timestamping on entities, relationships, and attributes) but without evolution constraints can be captured by $\mathsf{S5}_{\mathcal{ALCQI}}$ TBoxes.

The main contribution of this paper is to show that reasoning in $\mathsf{S5}_{\mathcal{ALCQI}}$ is decidable. We also pinpoint the exact computational complexity by showing 2-ExpTime completeness. Thus, adding the S5 *change modality* pushes the complexity of $\mathcal{ALCQI}$, which is ExpTime-complete, by one exponential. Our upper bound can be viewed as an extension of the decidability result for a simpler multi-dimensional DL, $\mathsf{S5}_{\mathcal{ALC}}$, [Gabbay *et al.*, 2003] which is not capable of capturing TER models. However, we had to develop completely new proof techniques as the decidability proof for $\mathsf{S5}_{\mathcal{ALC}}$ re-

*The research was supported by EU IST-2005-7603 FET Project Tones and by NSERC.

lies on the ability to duplicate points in a model, which is impossible in the presence of $\mathsf{S5}_{\mathcal{ALCQI}}$'s number restrictions. Our lower bound applies also to $\mathsf{S5}_{\mathcal{ALC}}$, hence we show that this logic is also 2-EXPTIME-complete.

The paper is organized as follows. Section 2 introduces the logic $\mathsf{S5}_{\mathcal{ALCQI}}$. Section 3 shows that reasoning in $\mathsf{S5}_{\mathcal{ALCQI}}$ is decidable in 2-EXPTIME by first providing a tree abstraction of $\mathsf{S5}_{\mathcal{ALCQI}}$ interpretations and then presenting a 2-EXPTIME procedure that checks for the existence of such a tree abstraction. Section 4 illustrates how $\mathsf{S5}_{\mathcal{ALCQI}}$ is able to capture conceptual models with timestamping constraints. Full proofs and the 2-EXPTIME lower bound can be found in the full version of this paper [Artale *et al.*, 2006].

2 The Logic $\mathsf{S5}_{\mathcal{ALCQI}}$

The logic $\mathsf{S5}_{\mathcal{ALCQI}}$ combines the modal logic S5 and the description logic $\mathcal{ALCQI}$ in a spirit similar to the multi-dimensional DLs [Gabbay *et al.*, 2003; Wolter and Zakharyaschev, 1999]. The syntax of $\mathsf{S5}_{\mathcal{ALCQI}}$ is defined as follows. Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be disjoint and countably infinite sets of *concept names* and *role names*. We assume that $\mathsf{N_R}$ is partitioned into two countably infinite sets $\mathsf{N_{glo}}$ and $\mathsf{N_{loc}}$ of *global role names* and *local role names*. The set ROL of *roles* is defined as $\{r, r^-, \Diamond r, \Diamond r^-, \Box r, \Box r^- \mid r \in \mathsf{N_R}\}$. The set of concepts CON is defined inductively as follows: $\mathsf{N_C} \subseteq \mathsf{CON}$; if $C, D \in \mathsf{CON}$, $r \in \mathsf{ROL}$, and $n \in N$, then the following are also in CON: $\neg C$, $C \sqcap D$, $(\geqslant n\ r\ C)$, and $\Diamond C$. A *TBox* is a finite set of *general concept inclusions (GCIs)* $C \sqsubseteq D$ with $C, D \in \mathsf{CON}$.

An $\mathsf{S5}_{\mathcal{ALCQI}}$*-interpretation* $\mathfrak{I}$ is a pair $(W, \mathcal{I})$ with W a non-empty set of *worlds* and $\mathcal{I}$ a function assigning an $\mathcal{ALCQI}$*-interpretation* $\mathcal{I}(w) = (\Delta, \cdot^{\mathcal{I},w})$ to each $w \in W$, with Δ a fixed non-empty set called the *domain* and $\cdot^{\mathcal{I},w}$ a function mapping each $A \in \mathsf{N_C}$ to a subset $A^{\mathcal{I},w} \subseteq \Delta$ and each $r \in \mathsf{N_R}$ to a relation $r^{\mathcal{I},w} \subseteq \Delta \times \Delta$; for $r \in \mathsf{N_{glo}}$ we additionally require $r^{\mathcal{I},w} = r^{\mathcal{I},v}$ for all $w, v \in W$. We extend the mapping $\cdot^{\mathcal{I},w}$ to complex roles and concepts as follows:

$$
\begin{aligned}
(r^-)^{\mathcal{I},w} &:= \{(y,x) \in \Delta \times \Delta \mid (x,y) \in r^{\mathcal{I},w}\}\\
(\Diamond r)^{\mathcal{I},w} &:= \{(x,y) \in \Delta \times \Delta \mid \exists v \in W : (x,y) \in r^{\mathcal{I},v}\}\\
(\Box r)^{\mathcal{I},w} &:= \{(x,y) \in \Delta \times \Delta \mid \forall v \in W : (x,y) \in r^{\mathcal{I},v}\}\\
(\neg C)^{\mathcal{I},w} &:= \Delta \setminus C^{\mathcal{I},w}\\
(C \sqcap D)^{\mathcal{I},w} &:= C^{\mathcal{I},w} \cap D^{\mathcal{I},w}\\
(\geqslant n\ r\ C)^{\mathcal{I},w} &:= \{x \in \Delta \mid \sharp\{y \in \Delta \mid (x,y) \in r^{\mathcal{I},w} \text{ and } y \in C^{\mathcal{I},w}\} \geq n\}\\
(\Diamond C)^{\mathcal{I},w} &:= \{x \in \Delta \mid \exists v \in W : x \in C^{\mathcal{I},v}\}
\end{aligned}
$$

An $\mathsf{S5}_{\mathcal{ALCQI}}$-interpretation $\mathfrak{I} = (W, \mathcal{I})$ is a *model* of a TBox $\mathcal{T}$ iff it satisfies $C^{\mathcal{I},w} \subseteq D^{\mathcal{I},w}$ for all $C \sqsubseteq D \in \mathcal{T}$ and $w \in W$. It is a *model* of a concept C if $C^{\mathcal{I},w} \neq \emptyset$ for some $w \in W$. A concept C is *satisfiable* w.r.t. a TBox $\mathcal{T}$ if there exists a common model of C and $\mathcal{T}$. Note that when $\mathsf{S5}_{\mathcal{ALCQI}}$ is considered a temporal description logic, the elements of W correspond to *time points*. We do not distinguish between global and local *concept* names because a concept name A can easily be enforced to be global using the GCI $A \sqsubseteq \Box A$.

The concept constructors $C \sqcup D$, $\exists r.C$, $\forall r.C$, $(\leqslant n\ r\ C)$, $(= n\ r\ C)$, $\Box C$, $\top$, and $\bot$ are defined as abbreviations in the usual way. For roles, we allow only single applications of S5 modalities and inverse. It is easily seen that roles obtained by nesting modal operators and inverse in an arbitrary way can be converted into an equivalent role in our simple form: multiple temporal operators are absorbed and inverses commute with temporal operators.

The fragment $\mathsf{S5}_{\mathcal{ALC}}$ of $\mathsf{S5}_{\mathcal{ALCQI}}$ is obtained by allowing only roles of the form r, $\Diamond r$, and $\Box r$, and replacing the concept constructor $(\geqslant n\ r\ C)$ by $\exists r.C$. We note that neither $\mathsf{S5}_{\mathcal{ALC}}$ nor $\mathsf{S5}_{\mathcal{ALCQI}}$ enjoys the finite model property: there are concepts and TBoxes that are only satisfiable in models with both an infinite set of worlds W and an infinite domain Δ. An example of this phenomenon is the concept $\neg C$ and the TBox $\{\neg C \sqsubseteq \Diamond C,\ C \sqsubseteq \exists r.\neg C,\ \neg C \sqsubseteq \forall r.\neg C\}$, with $r \in \mathsf{N_{glo}}$.

3 Reasoning in $\mathsf{S5}_{\mathcal{ALCQI}}$

We show that in $\mathsf{S5}_{\mathcal{ALCQI}}$, satisfiability w.r.t. TBoxes is decidable in 2-EXPTIME. For simplicity, throughout this section we assume that only local role names are used. This can be done w.l.o.g. as global role names can be simulated by $\Box r$, for r a fresh local role name. Let C_0 and $\mathcal{T}$ be a concept and a TBox whose satisfiability is to be decided. We introduce the following notation. For roles r, we use $\mathsf{Inv}(r)$ to denote r^- if $r \in \mathsf{N_R}$, s if $r = s^-$, $\Diamond\mathsf{Inv}(s)$ if $r = \Diamond s$, and $\Box\mathsf{Inv}(s)$ if $r = \Box s$. We use $\mathsf{rol}(C_0, \mathcal{T})$ to denote the smallest set that contains all sub-roles used in C_0 and $\mathcal{T}$ and is closed under Inv. We use $\mathsf{cl}(C_0, \mathcal{T})$ to denote the smallest set containing all sub-concepts appearing in C_0 and $\mathcal{T}$ that is closed under negation: if $C \in \mathsf{cl}(C_0, \mathcal{T})$ and "$\neg$" is not the top level operator in C, then $\neg C \in \mathsf{cl}(C_0, \mathcal{T})$.

In the rest of this section we devise tree abstractions of models of C_0 and $\mathcal{T}$ which we call $(C_0, \mathcal{T})$*-trees*. In the subsequent section, we then show how to construct looping tree automata that accept the $(C_0, \mathcal{T})$-trees and thus reduce satisfiability in $\mathsf{S5}_{\mathcal{ALCQI}}$ to the emptiness problem of looping tree automata, yielding decidability of $\mathsf{S5}_{\mathcal{ALCQI}}$.

3.1 Tree Abstractions of $\mathsf{S5}_{\mathcal{ALCQI}}$ models

Intuitively, for a $(C_0, \mathcal{T})$-tree τ that abstracts a model $\mathfrak{I}$ of C_0 and $\mathcal{T}$, the root node of τ corresponds to an object x in $\mathfrak{I}$ that realizes C_0. Successors of the root in τ correspond to objects in $\mathfrak{I}$ that can be reached from x by traversing a role in some S5 world. Similarly, further nodes in τ correspond to objects of $\mathfrak{I}$ reachable from x by traversing multiple roles. To describe the concept and role interpretations of $\mathfrak{I}$ in its abstraction τ, we decorate the nodes of τ with *extended quasistates* as introduced in Definition 3 below. Extended quasistates are defined in terms of types and quasistates, which we introduce first. Intuitively, a type describes the concept memberships of a domain element $x \in \Delta$ in a single S5 world.

Definition 1 (Type)**.** A *type* t *for* $C_0, \mathcal{T}$ is a subset of $\mathsf{cl}(C_0, \mathcal{T})$ such that

$$
\begin{aligned}
&\neg C \in t \text{ iff } C \notin t && \text{for } \neg C \in \mathsf{cl}(C_0, \mathcal{T})\\
&C \sqcap D \in t \text{ iff } C \in t \text{ and } D \in t && \text{for } C \sqcap D \in \mathsf{cl}(C_0, \mathcal{T})\\
&D \in t \text{ if } C \in t && \text{for } C \sqsubseteq D \in \mathcal{T}
\end{aligned}
$$

We use $\mathsf{tp}(C_0, \mathcal{T})$ to denote the set of all types for C_0 and $\mathcal{T}$. To describe the concept memberships of a domain element in *all* S5 worlds, we use quasistates:

Definition 2 (Quasistate)**.** Let W be a set and $f : W \to \mathsf{tp}(C_0, \mathcal{T})$ a function such that for all $w \in W$ we have:

$$\diamond C \in f(w) \text{ iff } C \in f(v) \text{ for some } v \in W.$$

We call the pair (W, f) a *quasistate witness* and the set $\{f(v) \mid v \in W\}$ a *quasistate*.

To check whether a set of types $\{t_1, \ldots, t_n\}$ forms a valid quasistate, we can simply check whether the pair (W, f), with $W = \{t_1, \ldots, t_n\}$ and f the identity function, is a quasistate witness. Note, however, that each quasistate has many witnesses.

Quasistates only abstract concept membership of a particular object in all worlds. To capture the role structure relating objects adjacent in *some* S5 world in a given $\mathsf{S5}_{\mathcal{ALCQI}}$ model, we develop the notion of a *extended quasistate*. Ultimately, in the desired tree abstraction these two objects turn into a parent and a child nodes; the child is then labeled by the extended quasistate in question. Note the similarity to handling inverse roles using the *double blocking* technique used in tableau algorithms for $\mathcal{ALCQI}$ [Horrocks *et al.*, 1999].

Definition 3 (Extended Quasistate)**.** Let W be a set, (W, f) and (W, g) quasistate witnesses, and $h : W \to \mathsf{rol}(C_0, \mathcal{T}) \cup \{\Box\varepsilon\}$ for $\varepsilon \notin \mathsf{rol}(C_0, \mathcal{T})$ such that, for every $r \in \mathsf{N_R} \cup \{s^- \mid s \in \mathsf{N_R}\}$:

1. if $h(w) = \diamond r$, for some $w \in W$, then $h(v) = r$, for some $v \in W$;
2. if $h(w) = r$, for some $w \in W$, then either $h(v) = \diamond r$ or $h(v) = r$, for all $v \in W$;
3. it is not the case that $h(w) = r$ for all $w \in W$;
4. if $h(w) = \Box r$, for some $w \in W$, then $h(v) = \Box r$, for all $v \in W$.

We call (W, f, g, h) an *extended quasistate witness* and the set of triples $Q(W, f, g, h) = \{(f(v), g(v), h(v)) \mid v \in W\}$ an *extended quasistate*. Elements of $Q(W, f, g, h)$ are called *extended types* and $\mathsf{etp}(C_0, \mathcal{T})$ denotes the set of all extended types for C_0 and $\mathcal{T}$. We say that $Q(W, f, g, h)$ *realizes a concept* C if $C \in f(w)$ for some $w \in W$; we say that $Q(W, f, g, h)$ *is root* if $h(w) = \Box\varepsilon$ for all $w \in W$.

Intuitively, given a node labeled with the extended quasistate witness (W, f, g, h), the quasistate witness (W, f) describes the node which is labeled with the extended quasistate, (W, g) describes the predecessor of this node, and h describes the role connections between the two nodes. Conditions 1 to 4 ensure that the mapping h assigns roles in a way that respects the semantics of modal operators. To fully understand these conditions, note that we assume an ordering $\diamond r \leq r \leq \Box r$ between roles which allows us to use a single role in the extended quasistate to capture all the *implied* roles. The dummy role ε is intended only for use with the root object, which does not have a predecessor.

We now introduce the concept of a *matching successor*. The main difficulty is to properly capture the effects of *qualified number restrictions* $(\geqslant n\ r\ C)$ which constrain the possible combinations of extended quasistates in $(C_0, \mathcal{T})$-trees: the extended quasistates assigned to children nodes must satisfy the qualified number restrictions of the parent node.

Definition 4 (Matching Successor)**.** Let W and Γ be sets, $x \notin \Gamma$, and let e be a function mapping each $y \in \Gamma \cup \{x\}$ to an extended quasistate witness (W, f_y, g_y, h_y) such that $g_y = f_x$ for all $y \in \Gamma$. We call (W, Γ, x, e) a *matching successor witness* if for all $w \in W$:

1. if $(\geqslant n\ r\ C) \in f_x(w)$ and $C \notin g_x(w)$ or $\mathsf{Inv}(r) \not\leq h_x(w)$ then $|\{y \in \Gamma \mid r \leq h_y(w), C \in f_y(w)\}| \geq n$,
2. if $(\geqslant n\ r\ C) \in f_x(w)$, then $|\{y \in \Gamma \mid r \leq h_y(w), C \in f_y(w)\}| \geq n - 1$,
3. if $(\geqslant n\ r\ C) \in \mathsf{cl}(\mathcal{T}, C_0)$, $C \in g_x(w)$, and $\mathsf{Inv}(r) \leq h_x(w)$, and $|\{y \in \Gamma \mid r \leq h_y(w), C \in f_y(w)\}| \geq n - 1$ then $(\geqslant n\ r\ C) \in f_x(w)$,
4. if $(\geqslant n\ r\ C) \in \mathsf{cl}(\mathcal{T}, C_0)$ and $|\{y \in \Gamma \mid r \leq h_y(w), C \in f_y(w)\}| \geq n$ then $(\geqslant n\ r\ C) \in f_x(w)$.

The pair $(Q(W, f_x, g_x, h_x), \{Q(W, f_y, g_y, h_y) \mid y \in \Gamma\})$ is called a *matching successor*,.

We say that two matching successor witnesses are *equivalent* if they define the same matching successor.

The intuition behind this definition is as follows: the object x stands for a parent node (described by f_x) and the set of objects Γ for all its children (described by f_y). The extended quasistates are chosen in a consistent way w.r.t. the information that is represented twice: the parent part of the extended quasistates labeling the children matches the quasistate attached to the parent itself (i.e., $g_y = f_x$ for all $y \in \Gamma$). A *matching successor witness* is then a witness such that the extended quasistates attached to x and to all elements of Γ can be used to build a part of a model of C_0 and $\mathcal{T}$ without violating any qualifying number restrictions. Also, the domain of such a model is eventually built from the objects $\{x\} \cup \Gamma$. As already mentioned, matching successors are the most crucial ingredient to the definition of $(C_0, \mathcal{T})$-trees.

Definition 5 ($(C_0, \mathcal{T})$-tree)**.** Let $\tau = (N, E, G, n_0)$ be a tuple such that (N, E) is a tree with root $n_0 \in N$ and G a mapping of τ's nodes to extended quasistates. Then τ is a $(C_0, \mathcal{T})$*-tree* if:

1. $G(n_0)$ realizes C_0;
2. $G(n_0)$ is root;
3. for all $n \in N$, the pair $(G(n), \{G(m) \mid (n, m) \in E\})$ is a matching successor.

Note that the *matching successor witnesses* induced by the matching successors consisting of the extended quasistates associated with a node and its children in the $(C_0, \mathcal{T})$-tree do not necessarily share the same set W. This poses a difficulty when showing that $(C_0, \mathcal{T})$-trees are proper abstractions of models of C_0 and $\mathcal{T}$: when we want to convert such a tree into an interpretation, we need to decide on a common set of worlds W. This difficulty can be overcome using the following lemma, which shows that we can assume that all extended quasistate witnesses are based on sets of world W of identical cardinality.

Lemma 6 (Compatible Matching Successor)**.** *There exists an infinite cardinal α such that the following holds: for every*

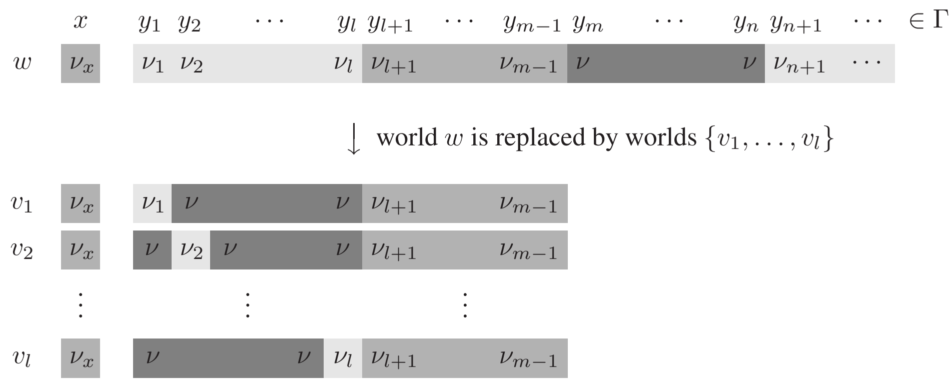

Figure 1: Reducing the size of a Matching Successor Witness.

matching successor witness (W, Γ, x, e)*, there is a matching successor witness* (W', Γ, x, e') *such that*

- (W, Γ, x, e) *and* (W', Γ, x, e') *define the same matching successor and;*
- *for all* $y \in \Gamma \cup \{x\}$ *and all extended types* $\nu \in Q(W', f'_y, g'_y, h'_y)$*, we have:*

$$|\{w \in W' \mid (f'_y(w), g'_y(w), h'_y(w)) = \nu\}| = \alpha.$$

Intuitively, the Lemma is proved by replicating elements of W a sufficient number of times. When all extended quasistate witnesses for a $(C_0, \mathcal{T})$-tree are based on a set of worlds of identical cardinality, we can connect these extended quasistate witnesses to a model by simply permuting the set W in an appropriate way. In this way, we can prove the difficult right-to-left direction of the following theorem.

Theorem 7. *C_0 is satisfiable w.r.t. $\mathcal{T}$ iff a $(C_0, \mathcal{T})$-tree exists.*

3.2 Decidability and Complexity of $\mathsf{S5}_{\mathcal{ALCQI}}$

We now develop an effective procedure to check whether there exists a $(C_0, \mathcal{T})$-tree for a given concept C_0 and TBox $\mathcal{T}$. We also show that the procedure runs in 2-ExpTime. Since it is easy to see that the number of all possible *matching successors* (q, Q) is 3-exponential, we cannot simply generate all of them and check whether they give rise to a $(C_0, \mathcal{T})$-tree. Instead, we start by showing that if a $(C_0, \mathcal{T})$-tree exists, then there is one with *slim matching successors witnesses* only, i.e., all matching successors in this tree are witnessed by matching successor witnesses whose size is at most exponential. In the rest of the paper, let $\mathsf{max}_{C_0,\mathcal{T}} = \sum_{(\geqslant m\ r\ C) \in \mathsf{cl}(C_0,\mathcal{T})} m$, and $n = |\mathsf{cl}(C_0, \mathcal{T})|$.

Lemma 8. *Let (W, Γ, x, e) be a matching successor witness for a matching successor (q, Q). Then there is a $Q' \subseteq Q$ and a matching successor witness (W', Γ', x, e') for (q, Q') such that: $|\Gamma'| \leq n \cdot 2^{2n} \cdot (\mathsf{max}_{C_0,\mathcal{T}} + 1)$, and $|W'| \leq (1 + |\Gamma'|) \cdot n \cdot 2^{2n}$.*

We call a matching successor witness *slim* if it satisfies the cardinality bounds given in the above Lemma. We call a matching successor *slim* if it has a slim matching successor witness.

To prove Lemma 8, we need to construct the required slim matching successor witness (W', Γ', x, e') from (W, Γ, x, e). To this end, we choose a set $\Gamma' \subseteq \Gamma$ and a function μ that associates an extended type $\mu(y)$ with every $y \in \Gamma'$ such that

- $\mu(y) \in Q(W, f_y, g_y, h_y)$ and
- if $\nu \in Q(W, f_z, g_z, h_z)$ for some $z \in \Gamma \setminus \Gamma'$ then we have $|\{y \in \Gamma' \mid \mu(y) = \nu\}| = \mathsf{max}_{C_0,\mathcal{T}} + 1$.

The function μ tells us which objects in Γ' can, in every particular world w, be used to fulfill number restrictions that have been originally fulfilled by (extended types of) objects in $\Gamma \setminus \Gamma'$.

The set Γ', in turn, is the basis to constructing a *slim witness* as it can always be chosen in a way such that $|\Gamma'| \leq |\mathsf{etp}(C_0, \mathcal{T})| \cdot (\mathsf{max}_{C_0,\mathcal{T}} + 1) \leq n \cdot 2^n \cdot (\mathsf{max}_{C_0,\mathcal{T}} + 1)$. Finally, for a witness in which $|\Gamma|$ is bounded as above, we can simply eliminate *superfluous worlds* of W to obtain a slim witness. This can be done by keeping at most $|\mathsf{etp}(C_0, \mathcal{T})| \leq n \cdot 2^{2n}$ worlds for each element of $\{x\} \cup \Gamma'$; those worlds can be chosen from W independently.

The crucial step of the actual construction is illustrated in Figure 1: consider a particular world w. In the original witness the number restrictions in the parent object are fulfilled, e.g., by the objects $y_{l+1}, \ldots, y_n$ with $y_m, \ldots, y_n$ falling outside of the set Γ'. Assume first, for simplicity, that the objects $y_m, \ldots, y_n$ have been assigned a common extended type ν. We then pick objects $y_1, \ldots, y_l \in \Gamma$ such that $\mu(y_1) = \cdots = \mu(y_l) = \nu$. Since $n - m + 1 \leq \mathsf{max}_{C_0,\mathcal{T}}$ we can always find $l \leq \mathsf{max}_{C_0,\mathcal{T}} + 1$ of such objects in Γ' such that $l = n - m + 2$; follows from the definition of Γ'. Thus we can transform the old witness to a new one as depicted in Figure 1. Whenever more than one extended type is associated with $y_m, \ldots, y_n$ in the original witness, we simply pick objects in Γ' with an appropriate matching μ value and proceed similarly to the above example. To construct a slim matching successor witness we fix the set Γ' and apply this transformation to all $w \in W$ independently. Note that the transformation preserves *quasistates* associated with all objects in Γ' (hence all $\mathsf{S5}$ modalities are preserved) and that all number restrictions are met.

Lemma 9. *There is a procedure that runs in 2-*ExpTime *to* *generate all slim matching successors.*

We simply use a brute-force approach to enumerate all candidates for slim matching successor witnesses up to exponentially sized Γ and W and test for satisfaction of the conditions in Definition 4.

As the next step, we show that whenever a $(C_0, \mathcal{T})$-tree exists, then there is a $(C_0, \mathcal{T})$-tree constructed solely from slim matching successors, i.e., $(G(n), \{G(m) \mid (n, m) \in E\})$ is a slim matching successor for all $n \in N$. We call such a $(C_0, \mathcal{T})$-tree *slim*.

Lemma 10. *For any C_0 and $\mathcal{T}$, a slim $(C_0, \mathcal{T})$-tree exists whenever a $(C_0, \mathcal{T})$-tree exists.*

Since the children in a slim matching successor are a subset of the children in the original matching successor, it is easy to convert an arbitrary $(C_0, \mathcal{T})$-tree into a slim one.

Finally, to check whether a slim $(C_0, \mathcal{T})$-tree exists we define a looping tree automaton $\mathcal{A}_{C_0,\mathcal{T}}$ that accepts exactly the slim $(C_0, \mathcal{T})$-trees. To check satisfiability of C_0 w.r.t. $\mathcal{T}$, it then suffices to check whether this looping automaton accepts at least one such tree. This observation yields a 2-ExpTime decision procedure for satisfiability in $\mathsf{S5}_{\mathcal{ALCQI}}$ as the emptiness problem for looping tree automata is decidable in time linear in the size of the automaton [Vardi and Wolper, 1986].

We use *extended quasistates* as states of the automaton $\mathcal{A}_{C_0,\mathcal{T}}$ and *slim matching successors* to define the transition relation. Since $C_0, \mathcal{T}$-trees do not have a constant branching degree, we use *amorphous* looping automata which are similar to the automata model introduced in [Kupferman and Vardi, 2001] except that in our case the branching degree is bounded and thus the transition relation can be represented finitely in a trivial way.

Definition 11 (Looping Tree Automaton)**.** A *looping tree automaton $\mathcal{A} = (Q, M, I, \delta)$* for an M-labeled tree is defined by a set Q of states, an alphabet M, a set $I \subseteq Q$ of initial states, and a transition relation $\delta \subseteq Q \times M \times 2^Q$.

Let $\mathfrak{T} = (N, E, \ell, r)$ be a tree with root $r \in N$ and labeling function $\ell : N \to M$. A *run* of $\mathcal{A}$ on $\mathfrak{T}$ is a mapping $\gamma : N \to Q$ such that $\gamma(r) \in I$ and $(\gamma(\alpha), \ell(\alpha), \{\gamma(\beta) \mid (\alpha, \beta) \in E\}) \in \delta$ for all $\alpha \in N$. A looping automaton $\mathcal{A}$ *accepts* those labeled trees $\mathfrak{T}$ for which there exists a run of $\mathcal{A}$ on $\mathfrak{T}$.

We construct an automaton for C_0 and $\mathcal{T}$ as follows.

Definition 12. Let C_0 be a concept and $\mathcal{T}$ an $\mathsf{S5}_{\mathcal{ALCQI}}$ TBox. The looping automaton $\mathcal{A}_{C_0,\mathcal{T}} = (Q, M, I, \delta)$ is defined by setting $M = Q = \mathsf{etp}(C_0, \mathcal{T})$, $I := \{q \in Q \mid q$ realizes C_0 and q is root$\}$, and δ to the set of those tuples $(q, q, \overline{Q})$ such that $\overline{Q} \in 2^Q$ and $(q, \overline{Q})$ is a slim matching successor for C_0 and $\mathcal{T}$.

The following Lemma states that the obtained looping automata behaves as expected.

Lemma 13. *τ is a slim $(C_0, \mathcal{T})$-tree iff τ is accepted by $\mathcal{A}_{C_0,\mathcal{T}}$.*

It is easily seen that there are at most 2-Exp many extended quasistates and thus $\mathcal{A}_{C_0,\mathcal{T}}$ has at most 2-Exp many states. To construct the transition function of the automaton, we need to construct all slim matching successors which can be done in 2-ExpTime by Lemma 9. Since emptiness of looping automata can be checked in polynomial time, the overall decision procedure for satisfiability in $\mathsf{S5}_{\mathcal{ALCQI}}$ runs in 2-ExpTime. This holds regardless of whether numbers inside number restrictions are coded in unary or in binary.

Theorem 14. *Satisfiability in $\mathsf{S5}_{\mathcal{ALCQI}}$ w.r.t TBoxes is decidable and 2-ExpTime-complete.*

The lower bound in Theorem 14 is obtained by reducing the word problem of exponentially space-bounded, alternating Turing machines. Since the reduction does not use inverse role and qualifying number restrictions, we also obtain a 2-ExpTime lower bound for satisfiability on $\mathsf{S5}_{\mathcal{ALC}}$.

Corollary 15. *Satisfiability in $\mathsf{S5}_{\mathcal{ALC}}$ w.r.t TBoxes is decidable and 2-ExpTime-complete.*

4 Capturing Conceptual Schemata

It is known that the TDL $\mathcal{ALCQI}_{\mathcal{US}}$ is able to capture the temporal conceptual model $\mathcal{ER}_{VT}$, a TER model that supports timestamping and evolution constraints, IS-A links, disjointness, covering, and participation constraints [Artale *et al.*, 2003]. In $\mathcal{ER}_{VT}$, timestamping is implemented using a marking approach as sketched in the introduction. The translation of atemporal constructs is similar to the one using $\mathcal{ALCQI}_{\mathcal{US}}$; see [Artale *et al.*, 2003] for full details and examples. In the following we briefly recall the translation of atemporal constructs and then show that $\mathsf{S5}_{\mathcal{ALCQI}}$ is sufficient to capture the fragment of $\mathcal{ER}_{VT}$ that has timestamping as the only temporal construct.

When translating $\mathcal{ER}_{VT}$ to TDLs, entities E—denoting sets of abstract objects—are mapped into concept names A_E and attributes P—denoting functions associating mandatory concrete properties of entities—are mapped into roles names r_P interpreted as total functions, which is enforced by the GCI $\top \sqsubseteq (= 1\ r_P\ \top)$. In $\mathsf{S5}_{\mathcal{ALCQI}}$, unrestricted entities and attributes need no special treatment. Properties of snapshot or temporary entities and attributes are captured as follows:

$A_E \sqsubseteq \Box A_E$	*snapshot entity*
$A_E \sqsubseteq \Diamond\neg A_E$	*temporary entity*
$A_E \sqsubseteq \exists\Box r_P.\top$	*snapshot attribute*
$A_E \sqsubseteq \forall\Box r_P.\bot$	*temporary attribute*

Relationships—n-ary relations between abstract objects—are translated using reification: each n-ary relationship R is translated into a concept name A_R with n *global* role names $r_1, \ldots, r_n$. Intuitively, for each instance $x \in A_R^{\mathcal{I},w}$, the tuple $(y_1, \ldots, y_n)$ with $(x, y_i) \in r_i^{\mathcal{I},w}$ is a tuple in the relationship R at a time point w. To ensure that every instance of A_R gives rise to a unique tuple in R, we use GCIs $\top \sqsubseteq (= 1\ r_i\ \top)$, for $1 \le i \le n$. To capture *snapshot relationships*, we assert $A_R \sqsubseteq \Box A_R$, while for *temporary relationships*, we assert $A_R \sqsubseteq \Diamond\neg A_R$ in the TBox.

Note that the latter GCIs do not fully capture temporary relationships. As an example, consider the interpretation $\mathfrak{I} = (\{w_1, w_2\}, \mathcal{I})$, with $\Delta = \{a, a', b, c\}$, $A_R^{\mathcal{I},w_1} = \{a\}$, $A_R^{\mathcal{I},w_2} = \{a'\}$, $r_1^{\mathcal{I},w_1} = \{(a, b)\}$, $r_2^{\mathcal{I},w_1} = \{(a, c)\}$, $r_1^{\mathcal{I},w_2} = \{(a', b)\}$, and $r_2^{\mathcal{I},w_2} = \{(a', c)\}$. Although the GCI

$A_R \sqsubseteq \Diamond\neg A_R$ (expressing temporary relationships) is satisfied, (b, c) is constantly in the temporary relationship R. This is due to a mismatch between the models of an $\mathcal{ER}_{VT}$ schema and the models of its translation into $\mathsf{S5}_{\mathcal{ALCQI}}$. In particular, in models of $\mathcal{ER}_{VT}$, tuples belonging to relationships are unique while in models of the reified translation there may be two *distinct* objects connected through the global roles r_i to the *same* objects, thus representing the same tuple (e.g. the objects a, a' in the above interpretation). Then, $\mathsf{S5}_{\mathcal{ALCQI}}$ models in which the above situation occurs do not directly correspond to any $\mathcal{ER}_{VT}$ model. Similarly to [Calvanese *et al.*, 1999], however, it is possible to show that: *(i)* there are so called *safe* models of $\mathsf{S5}_{\mathcal{ALCQI}}$ that are in one-to-one correspondence with $\mathcal{ER}_{VT}$ models, and *(ii)* every satisfiable $\mathsf{S5}_{\mathcal{ALCQI}}$ concept is also satisfied in a safe model. When reasoning about $\mathcal{ER}_{VT}$ schemas, we can thus safely ignore non-safe models. An $\mathsf{S5}_{\mathcal{ALCQI}}$ interpretation $\mathfrak{I} = (W, \mathcal{I})$ is *safe* for an $\mathcal{ER}_{VT}$ schema if, for every n-ary relationship R reified with the global functional roles r_i, and every $w \in W$, we have the following:

$$\forall x, y, x_1, \ldots, x_n \in \Delta : \neg((x, x_1) \in r_1^{\mathcal{I},w} \wedge (y, x_1) \in r_1^{\mathcal{I},w} \wedge \ldots \wedge (x, x_n) \in r_n^{\mathcal{I},w} \wedge (y, x_n) \in r_n^{\mathcal{I},w}).$$

It is not hard to see that: *(1)* the model in the example above is not safe, and *(2)* given a safe model, the above GCIs correctly capture the temporal behavior of relationships.

5 Conclusions

This work introduces the modal description logic $\mathsf{S5}_{\mathcal{ALCQI}}$ as a logic for representing and reasoning in temporal conceptual models with timestamping constraints. A novel technique is used to show decidability and 2-EXPTIME-completeness for $\mathsf{S5}_{\mathcal{ALCQI}}$. This is also the first decidability result that allows reasoning in temporal conceptual models with timestamping for entities, relationships, and attributes. Furthermore, reasoning on the less expressive logic $\mathsf{S5}_{\mathcal{ALC}}$ is also shown to be 2-EXPTIME-complete.

This paper leaves several interesting open problems for further investigation. The fine line separating the decidable TDLs from the undecidable ones is not fully explored: we plan to investigate further extensions of $\mathsf{S5}_{\mathcal{ALCQI}}$ that still enjoy decidability. Two natural candidates are $\mathsf{S5}_{\mathcal{ALCQI}}$ that allows, in addition to S5 modalities, an irreflexive $\Box$ (thus enabling statements about *everywhere else*) and $\mathsf{S5}_{\mathcal{ALCQI}}$ with temporalized axioms (enabling TBox statements to appear in scope of S5 operators). Another open issue concerns decidability and complexity of $\mathsf{S5}_{\mathcal{ALCQI}}$ in finite models.

On the knowledge representation side, we believe that a converse translation—from TER with full timestamping to $\mathsf{S5}_{\mathcal{ALCQI}}$—is also possible; this result would allow to fully characterize the complexity of reasoning in TER with timestamping. The limits of the expressive power of $\mathsf{S5}_{\mathcal{ALCQI}}$ w.r.t. various constraints that have appeared in literature on temporal models other than timestamping also remain to be determined.

References

[Artale and Franconi, 1999] A. Artale and E. Franconi. Temporal ER modeling with description logics. In *Proc. of the Int. Conference on Conceptual Modeling (ER'99)*, volume 1728 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999.

[Artale *et al.*, 2002] 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, *Proc. of the 8th Joint European Conference on Logics in Artificial Intelligence (JELIA-02)*, volume 2424 of *LNAI*, pages 98–110. Springer, 2002.

[Artale *et al.*, 2003] A. Artale, E. Franconi, and F. 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-Verlag, 2003.

[Artale *et al.*, 2006] A. Artale, C. Lutz, and D. Toman. A description logic of change. Technical report, LTCS-Report 05-06, Technical University Dresden, 2006. see http://lat.inf.tu-dresden.de/research/reports.html.

[Artale, 2004] A. Artale. Reasoning on temporal conceptual schemas with dynamic constraints. In *11th Int. Symposium on Temporal Representation and Reasoning (TIME04)*. IEEE Computer Society, 2004. Also in Proc. of DL'04.

[Calvanese *et al.*, 1998] D. Calvanese, M. Lenzerini, and D. Nardi. Description logics for conceptual data modeling. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*, pages 229–263. Kluwer, 1998.

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

[Chomicki and Toman, 2005] J. Chomicki and D. Toman. Temporal Databases. In M. Fischer, D. Gabbay, and L. Villa, editors, *Handbook of Temporal Reasoning in Artificial Intelligence*, pages 429–467. Elsevier *Foundations of Artificial Intelligence*, 2005.

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

[Horrocks *et al.*, 1999] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In *Proc. of LPAR'99*, number 1705 in LNAI, pages 161–180. Springer, 1999.

[Kupferman and Vardi, 2001] O. Kupferman and M. Y. Vardi. On bounded specifications. In *Proc. of the Int. Conference on Logic for Programming and Automated Reasoning (LPAR'01)*, LNAI, pages 24–38. Springer-Verlag, 2001.

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

[Vardi and Wolper, 1986] M. Y. Vardi and P. Wolper. Automata-theoretic techniques for modal logic of programs. *Journal of Computer and System Sciences*, 32:183–221, 1986.

[Wolter and Zakharyaschev, 1999] F. Wolter and M. Zakharyaschev. Modal descriptiopn logics: modalizing roles. *Fundamentae Informaticae*, 39:411–438, 1999.