A Description Logic of Change

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

Abstract

We combine the modal logic $\mathsf{S5}$ with the description logic (DL) $\mathcal{ALCQI}$. In this way, we obtain a multi-dimensional DL, $\mathsf{S5}_{\mathcal{ALCQI}}$, whose purpose is reasoning about change. $\mathsf{S5}_{\mathcal{ALCQI}}$ is capable of expressing that concepts and roles change over time, but cannot discriminate between changes in the past and in the future. Our main technical result is that $\mathsf{S5}_{\mathcal{ALCQI}}$ concept satisfiability with respect to terminologies of general concepts inclusions (GCIs) is decidable and 2-ExpTime-hard. We also provide a scenario based on temporal conceptual models with timestamping constraints in which the logic can be used.

1 Introduction

An important application of Temporal Description Logics (TDLs) is the representation of and reasoning about temporal conceptual models [1, 3, 4]. The general idea is that such models are translated into an appropriate TDL, and that TDL reasoning is then used to detect inconsistencies and implicit IS-A relations in the temporal model [2, 4, 6]. However, 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 witness of this problem is constituted by temporal entity-relationship (TER) models used to design temporal databases [8]. TERs are classical ER data models extended with two classes of constraints that model the temporal behavior of an application domain [13], i.e., *evolution constraints*—to govern *object migration*, i.e., the inability, possibility, and necessity of objects to change membership from one entity to another—and *timestamping*—to distinguish between temporal and atemporal components of a TER model.

In this paper, we are interested in devising a logic able to fully capture TER with timestamping constraints and study its computational properties. Indeed, timestamping are temporal constructs supported by almost all temporal conceptual models proposed in the literature [10, 11, 13, 14]. It has be implemented either by marking entities (i.e., classes), relationships and attributes as *snapshot* or *temporary*, or leaving them un-marked. Then, objects belong to a snapshot entity either never or at all times, no object may belong to a temporary entity at all times, and there are no (temporal) assumptions about instances of un-marked entities. The meaning of timestamps for relationships and attributes is analogous.

It has been observed in [4] that TER models with timestamping and evolution constraints can be translated into the TDL $\mathsf{DLR}_{\mathcal{US}}$. In the same paper, undecidability of $\mathsf{DLR}_{\mathcal{US}}$ was established. Later, it was shown in [1] that these computational problems are not due to the translation to description logics: even direct reasoning with the (less powerful) TER models is undecidable. There appear to be two ways around this problem. First, one can restrict timestamping. Indeed, it has been shown in [4] that giving up timestamping of both relationships and attributes (in the sense that all relationships and attributes are left un-marked) re-establishes decidability of $\mathsf{DLR}_{\mathcal{US}}$, and thus also of TER models with both restricted timestamping and evolution constraints. Second, one can allow for full timestamping but avoiding 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 modal $\mathsf{S5}$ with the standard DL $\mathcal{ALCQI}$. The $\mathsf{S5}$ modality can be applied to both concepts and roles but not in front of axioms—terminological axioms are interpreted globally. This logic may be viewed as a *description logic of change*, as it allows to state that concept and role memberships change in time, but does not allow to discriminate between changes in the past and changes in the 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 $\mathsf{S5}_{\mathcal{ALCQI}}$ terminologies. The main result of this paper is to show that reasoning in $\mathsf{S5}_{\mathcal{ALCQI}}$ is decidable and 2-ExpTime-hard. While a decidability result was obtained for the simpler modal DL, $\mathsf{S5}$-$\mathcal{ALC}$ [9], this is the first time that the more complex $\mathsf{S5}_{\mathcal{ALCQI}}$ is showed to have a decidable reasoning problem. Thus, dropping evolution constraints indeed recovers decidability. However, it is surprising that adding change, a rather weak form of temporality, pushes the complexity of $\mathcal{ALCQI}$ from ExpTime-complete to 2-ExpTime-hard.

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

The logic $\mathsf{S5}_{\mathcal{ALCQI}}$ is a combination of the epistemic modal logic $\mathsf{S5}$ and the description logic $\mathcal{ALCQI}$. It is similar in spirit to the multi-dimensional description logics proposed, e.g., in [9, 16]. 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 $\{r, r^-, \Diamond r, \Diamond r^-, \Box r, \Box r^-\}$, with $r \in \mathsf{N_R}$. The set of concepts CON is defined inductively: $\mathsf{N_C} \subseteq \mathsf{CON}$; if $C, D \in \mathsf{CON}$, $r \in \mathsf{ROL}$, and $n \in \mathbb{N}$, 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 $C, D \in \mathsf{CON}$.

The concept constructors $C \sqcup D$, $\exists r.C$, $(\leqslant n\ r\ C)$, $(= n\ r\ C)$, $\forall r.C$, $\Box C$, $\top$, and $\bot$ are defined as abbreviations in the usual way. Concerning roles, note that we allow only single applications of boxes and diamonds, while inverse is applicable only to role names. It is easily seen that any role obtained by nesting modal operators and inverse in an arbitrary way can be converted into an equivalent role in this restricted form: multiple temporal operators are absorbed and inverse commutes over temporal operators [5].

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 to each $w \in W$ an $\mathcal{ALCQI}$*-interpretation* $\mathcal{I}(w) = (\Delta, \cdot^{\mathcal{I},w})$, where the *domain* Δ is a non-empty set and $\cdot^{\mathcal{I},w}$ is 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 $r^{\mathcal{I},w} \subseteq \Delta \times \Delta$, such that if $r \in \mathsf{N_{glo}}$, then $r^{\mathcal{I},w} = r^{\mathcal{I},v}$ for all $w, v \in W$. We can 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 $w \in W$.

Note that $\mathsf{S5}_{\mathcal{ALCQI}}$ does not have the finite model property: there are concepts and TBoxes that are only satisfiable in models with both an infinite set of worlds and an infinite domain. For example, this is true for the concept $\neg C$ w.r.t. the TBox $\{\neg C \sqsubseteq \Diamond C,\ C \sqsubseteq \exists r.\neg C,\ \neg C \sqsubseteq \forall r.\neg C\}$, where $r \in \mathsf{N_{glo}}$.

3 Capturing Conceptual Schemas

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 and covering constraints, participation constraints [3]. In $\mathcal{ER}_{VT}$, timestamping is implemented using a marking approach as sketched in the introduction. Since the translation of atemporal constructs is similar to the one using $\mathcal{ALCQI}_{\mathcal{US}}$ the reader must refer to [3] for full details and examples. In the following, after recalling the translation of atemporal constructs, we 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 while attributes P—denoting functions associating mandatory concrete properties to entities—are mapped into roles names r_P that are enforced to be interpreted as total functions using GCIs: $\top \sqsubseteq (= 1\ r_P\ \top)$. In $\mathsf{S5}_{\mathcal{ALCQI}}$, un-marked entities and attributes need no special treatment, while entities and attributes being of type snapshot or temporary can be expressed 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—denoting n-ary relations between abstract objects—are translated by reification, i.e., 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 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 $1 \leq i \leq n$. Now, to capture *snapshot relationships*, we simply put $A_R \sqsubseteq \Box A_R$, while for *temporary relationships*, we put $A_R \sqsubseteq \Diamond\neg A_R$.

Note that, the latter GCI does 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 particulars, 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 r_i global roles to the *same* objects, thus representing the same tuple, e.g., the distinct objects a, a' in the above interpretation. Then, $\mathsf{S5}_{\mathcal{ALCQI}}$ models verifying the above situation do not correspond directly to an $\mathcal{ER}_{VT}$ model. However, similarly to [7], 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, *(ii)* every satisfiable $\mathsf{S5}_{\mathcal{ALCQI}}$ concept is also satisfied in a safe model. Since we are interested in reasoning about $\mathcal{ER}_{VT}$ schemas we can thus forget about 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, *(2)* given a safe model, the above GCI for temporary relationships correctly capture the safety property.

4 Decidability of Reasoning in $\mathsf{S5}_{\mathcal{ALCQI}}$

We show that the satisfiability problem is decidable for $\mathsf{S5}_{\mathcal{ALCQI}}$. For simplicity, throughout this section we assume that only local role names are used. This can be done w.l.o.g. since global role names can be simulated by $\Box r$, where r is a fresh local role name. To prove decidability, we start with devising tree abstractions of $\mathsf{S5}_{\mathcal{ALCQI}}$ models. Then we show how, given a concept C_0 and TBox $\mathcal{T}$, we can construct a looping tree automaton accepting exactly the tree abstractions of models of C_0 and $\mathcal{T}$.

Let C_0 and $\mathcal{T}$ be a concept and a TBox whose satisfiability is to be decided. We first 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 roles used in C_0 and $\mathcal{T}$, and that 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 $\mathcal{T}$ closed under negation, if $C \in \mathsf{cl}(C_0, \mathcal{T})$ and "$\neg$" is not the top level operator in C, then $\neg C \in \mathsf{cl}(C_0, \mathcal{T})$.

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

The goal of this section is to develop *tree abstractions* of general $\mathsf{S5}_{\mathcal{ALCQI}}$ models: for a given model $\mathfrak{I}$ of C_0 and $\mathcal{T}$, we develop a *tree abstraction* called $(C_0, \mathcal{T})$-tree. The root node corresponds to the object that realizes C_0 in $\mathfrak{I}$, descendants of the root correspond to further objects in $\mathfrak{I}$ that can be reached by traversing roles in *some* S5 world staring from the *root* object. For the abstractions to capture the essence of the $\mathsf{S5}_{\mathcal{ALCQI}}$ models, we need to attach additional information to the nodes of the tree and to constrain the parent-child relationships. To this end we develop the notions of an *extended quasistate* and *matching successor*. In the rest of this section we formalize the above intuition.

We first introduce types and quasistates. Intuitively, a type describes the concept memberships of a domain element $x \in \Delta$ in a single $\mathsf{S5}$ world.

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

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

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

In particular, quasistates capture constraints implied by the S5 modalities on concept membership of an object in all worlds. To check whether a set of types, $\{t_1, \ldots, t_n\}$, is a quasistate we 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.

To abstract the role structure of a model we define the notion of an *extended quasistate*. This abstraction is realized by a pair of quasistates in a $(C_0, \mathcal{T})$*-trees* and captures how two objects are related by a particular role.

Definition 3 (Extended Quasistate). Let W be a set, $f, g : W \to \mathsf{tp}(C_0, \mathcal{T})$, and $h : W \to \mathsf{rol}(C_0, \mathcal{T}) \cup \{\epsilon\}$ for $\epsilon \notin \mathsf{rol}(C_0, \mathcal{T})$ such that

1. (W, f) and (W, g) are quasistate witnesses;
2. if $\Diamond r = h(w)$ for some $w \in W$, then $r = h(v)$ for some $v \in W$;
3. if $r = h(w)$ for some $w \in W$, then $\Diamond r = h(v)$ or $r = h(v)$ for all $v \in W$;
4. it is not the case that $r = h(w)$ for all $w \in W$;
5. if $\Box r = h(w)$ for some $w \in W$, then $\Box r = h(v)$ 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*.

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: $h(w) = \Box\epsilon$ for all $w \in W$.

The extended quasistates therefore enforce S5 modalities when applied to roles. In thee tree-shaped model abstraction we also use extended quasistates capture the idea that each object has exactly one parent: by convention we assume that the object abstracted by f has a parent abstracted by g. The two objects are connected by a role r that is abstracted by h in those worlds w in which $h(w) \in \{r, \Box r\}$. For uniformity, we use the ϵ dummy role for the root object[1]. We define an ordering between modalities of a basic role r as $\Diamond r \leq r \leq \Box r$, this arrangement allows us to use a single role in the extended quasistate to capture all the *implied* modalities. It is again immediate to verify whether a set of triples forms an extended quasistate.

The last ingredient needed in the tree abstractions is the ability to properly capture the effects of *number restrictions*. These constraints, given an object in a model, impact what and how many other objects can be connected to this object via roles (and in which worlds). In the tree abstraction this yields restrictions on which extended quasistates can possibly be children of a given quasistate. This intuition is captured by the notion of a matching successor:

Definition 4 (Matching Successor). Let W and O be sets, o an element such that $o \notin O$; $f, g : (O \cup \{o\}) \to W \to \mathsf{tp}(C_0, \mathcal{T})$, and $h : (O \cup \{o\}) \to W \to \mathsf{rol}(C_0, \mathcal{T})$ functions such that $(W, f(p), g(p), h(p))$ is an extended quasistate witness for all $p \in O \cup \{o\}$ and $f(o) = g(p)$ for all $p \in O$. We call (W, O, o, f, g, h) a *matching successor witness* if for all $w \in W$:

1. if $(\geqslant n\ r\ C) \in f(o)(w)$ and $C \notin g(o)(w)$ or $\mathsf{Inv}(r) \not\leq h(o)(w)$ then $|\{p \in O \mid r \leq h(p)(w), C \in f(p)(w)\}| \geq n$,

2. if $(\geqslant n\ r\ C) \in f(o)(w)$, then $|\{p \in O \mid r \leq h(p)(w), C \in f(p)(w)\}| \geq n-1$,

3. if $(\geqslant n\ r\ C) \in \mathsf{cl}(\mathcal{T}, C_0)$, $C \in g(o)(w)$, and $\mathsf{Inv}(r) \leq h(o)(w)$, and $|\{p \in O \mid r \leq h(p)(w), C \in f(p)(w)\}| \geq n-1$ then $(\geqslant n\ r\ C) \in f(o)(w)$,

4. if $(\geqslant n\ r\ C) \in \mathsf{cl}(\mathcal{T}, C_0)$ and $|\{p \in O \mid r \leq h(p)(w), C \in f(p)(w)\}| \geq n$ then $(\geqslant n\ r\ C) \in f(o)(w)$,

We call the pair $(Q(W, f(o), g(o), h(o)), \{Q(W, f(p), g(p), h(p)) \mid p \in O\})$ a *matching successor*. We say that two matching successor witnesses are *equivalent* if they define the same matching successor.

---

[1]Note that each node in the abstraction is labeled by an extended quasistate describing the corresponding object, its parent, and the role that connects them. This arrangement is needed in order to keep track of how inverses interact with number restrictions and is similar to the so-called *double-blocking* technique.

The intuition behind this definition is as follows: the object o stands for the parent node and the set of objects O for all its children in the tree abstraction. Each of these objects is labeled by an extended quasistate in a consistent way (i.e., the parent parts of the extended quasistates labeling the children match the quasistate attached to the parent). A *matching successor witness* is then a witness that the extended quasistates attached to o and to all elements of O can potentially be used to build a part of a model of C_0 and $\mathcal{T}$ without violating any constraints in $\mathcal{T}$. Thus we can use matching successors to define the sought after tree shaped abstraction of models of C_0 and $\mathcal{T}$.

Definition 5 (($C_0, \mathcal{T}$)-tree). Let $T = (n_0, N, E)$ be an ordinal tree with root $n_0 \in N$ and G a mapping of T's nodes to extended quasistates. We say that T is a $(C_0, \mathcal{T})$-tree if:

1. C_0 is realized in $G(n_0)$,
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.

To be able to eventually construct a model from our abstraction we use the following lemma that allows us to *concatenate* matching successor witnesses along branches of such a tree abstraction:

Lemma 6. *Let (W, O, o, f, g, h) be a matching successor witness for (q, Q) and α an infinite cardinal. Then there is an equivalent matching successor witness (W', O, o, f', g', h') such that*

$$|\{w \in W' \mid (f'(p)(w), g'(p)(w), h'(p)(w)) \text{ is a constant triple}\}| = \alpha$$

for all $p \in O \cup \{o\}$.

This can always be achieved by replicating elements of W sufficiently many times. These witnesses are convenient as whenever the associated extended quasistate match they can be *plugged* one into the bottom of another just by permuting the set W.

The intuitions behind Definitions 4 and 5 is as follows: when given a model $\mathfrak{I}$ of C_0 and $\mathcal{T}$, we use the object $o \in \Delta$ that realizes C_0 in $\mathfrak{I}$ to define the extended quasistate for the root of our tree abstraction; we then collect all the extended quasistates for all successors of o in $\mathfrak{I}$ and make them children of the root quasistate. Repeating this construction yields a $(C_0, \mathcal{T})$-tree: the children of every node have been extracted from $\mathfrak{I}$ and thus must satisfy the conditions on matching successors (the witness comes from the model). Conversely, given a $(C_0, \mathcal{T})$-tree, we use the fact that for each matching successor, there must be a witness. We use these witnesses, with the help of Lemma 6, to construct a model by traversing the $(C_0, \mathcal{T})$-tree top down while concatenating the matching successor witnesses found along the branches. Thus the existence of a $(C_0, \mathcal{T})$-tree is equivalent to the existence of a $(C_0, \mathcal{T})$ model:

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

4.1.1 Decidability and Looping Tree Automata

To show decidability, we need to show existence of at least one $(C_0, \mathcal{T})$-tree. We proceed in two steps:

1. We need to determine whether (q, Q) is a *matching successor* for each extended quasistate q and set of extended quasistates Q.

2. We need to show that the matching successors can be arranged into a tree rooted by a node labeled by an extended quasistate realizing C_0.

Thus, we first need to define a procedure that constructs all matching successors given C_0 and $\mathcal{T}$. To show this, we need the following lemma, where $\max(C_0, \mathcal{T}) := \sum_{(\geqslant m\ r\ C) \in \mathsf{cl}(C_0, \mathcal{T})} m$, and $n = |\mathsf{cl}(C_0, \mathcal{T})|$.

Lemma 8. *Let (W, O, o, f, g, h) be a matching successor witness for a matching successor (q, Q). Then there is an equivalent matching successor witness (W', O', o, f', g', h') such that $|O'| \leq (\max(C_0, \mathcal{T})+1) \cdot 2^{n2^{2n}}$ and $|W'| \leq 2^{|O'| \cdot (n+1)}$.*

Figure 1 illustrates why the bounded matching successor witness required by

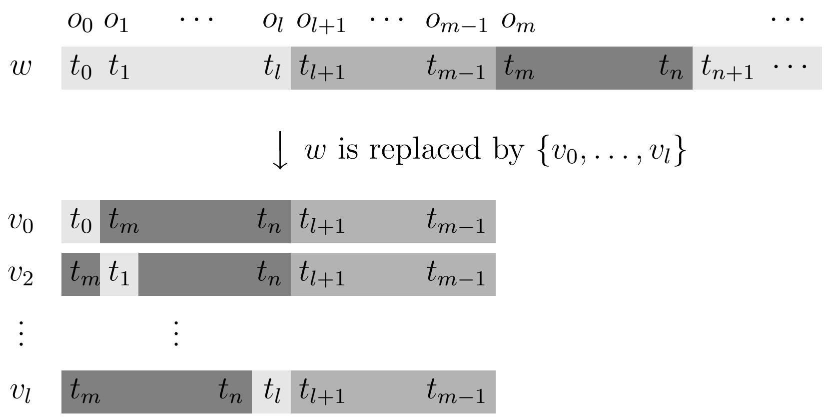

Figure 1: Reducing the size of a Matching Successor Witness.

Lemma 8 must exist: consider that the objects $o_0, \ldots, o_{n+1} \ldots \in O$ have been assigned the same extended quasistate by the current witness and that there are more than $\max(C_0, \mathcal{T}) + 1$ of such objects. Now consider a world w depicted on the top of the Figure. Assume that the (types of the) objects $o_{l+1}, \ldots, o_n$ are those needed to fulfill a number restriction in the (type of the) parent node in w. These must be *preserved* in the new witness after removing superfluous objects. The transformation is depicted in the lower half of the figure: we create a *set of worlds* $v_0, \ldots, v_l$ with the same parent label and such that all the types $t_{l+1}, \ldots, t_n$ are present in *every* of the new worlds. Hence the number restriction in the parent's type still holds. Moreover, the original type assigned to the objects $o_0, \ldots, o_l$ is preserved in *at least one* of the new worlds. This guarantees that, in the new witness, all the objects are still labeled by the same extended quasiworld. Thus the new witness is equivalent to the original one. Applying this procedure to every $w \in W$ and every set of objects labeled with the same extended quasiworld by the original witness removes all the superfluous objects that were assigned the same extended quasistate; hence the bound holds for the new witness. Thus we can test all matching successor candidates for witnesses up to the size defined in Lemma 8—in 4-ExpTime we can check whether a pair (q, Q) is a matching successor and there are at most 3-Exp of such candidates to test.

What remains is to check whether a $(C_0, \mathcal{T})$-tree can be constructed using these matching successors: to this end we define a looping tree automaton $\mathcal{A}_{C_0,\mathcal{T}}$ that accepts exactly the $C_0, \mathcal{T}$-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 tree. We show that this provides us with a 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 [15].

Intuitively, we use the *matching successors* to define the transition relation of $\mathcal{A}_{C_0,\mathcal{T}}$. Since in our case the trees do not have a constant branching degree, we adopt a variant of a looping automaton on amorphous trees [12] (except in our case we know the branching degree is bounded and thus the transition relation can be represented finitely in a trivial way).

Definition 9 (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 $\delta \subseteq Q \times M \times 2^Q$.

A *run* of $\mathcal{A}$ on an M-labeled tree $\mathfrak{T} = (r_\mathfrak{T}, N_\mathfrak{T}, E_\mathfrak{T})$ with a root $r_\mathfrak{T}$ is a mapping $\tau : N_\mathfrak{T} \to Q$ such that $\tau(r_\mathfrak{T}) \in I$ and $(\tau(\alpha), \mathfrak{T}(\alpha), \{\tau(\beta) \mid (\alpha, \beta) \in E_\mathfrak{T}\}) \in \delta$ for all $\alpha \in N_\mathfrak{T}$. A looping automaton $\mathcal{A}$ *accepts* those M-labeled trees $\mathfrak{T}$ for which there exists a run of $\mathcal{A}$ on $\mathfrak{T}$.

We construct an automaton from C_0 and $\mathcal{T}$ as follows:

Definition 10. Let C_0 be a concept and $\mathcal{T}$ a $\mathsf{S5}_{\mathcal{ALCQI}}$ TBox. We denote with $\mathsf{nl}(C_0, \mathcal{T})$ the set of all extended quasistates for C_0 and $\mathcal{T}$. A looping automaton $\mathcal{A}_{C_0,\mathcal{T}} = (Q, M, I, \delta)$ is defined by setting $M = Q = \mathsf{nl}(C_0, \mathcal{T})$, $I := \{q \in Q \mid q \text{ realizes } C_0,\ q \text{ is root}\}$, 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 matching successor for C_0 and $\mathcal{T}$.

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

Lemma 11. *$\mathfrak{T}$ is a $C_0, \mathcal{T}$-tree iff $\mathfrak{T}$ is accepted by $\mathcal{A}_{C_0,\mathcal{T}}$.*

The size of $\mathcal{A}_{C_0,\mathcal{T}}$ depends on the size of C_0 and $\mathcal{T}$ as there are only finitely many matching successors as shown in Lemma 8. To construct the transition function of the automaton, we also need to verify that the pair $(q, \overline{Q})$ is a matching successor. From Lemma 8, this can be done in 4-ExpTime. , we obtain that the proposed algorithm for satisfiability in $\mathsf{S5}_{\mathcal{ALCQI}}$ runs in 4-ExpTime. A lower bound can be established by reducing the word problem of exponentially space-bounded, alternating Turing machines [5].

Theorem 12. *Satisfiability in $\mathsf{S5}_{\mathcal{ALCQI}}$ is decidable and 2-*ExpTime*-hard.*

This result holds regardless of whether numbers inside number restrictions are coded in unary or in binary.

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 only. $\mathsf{S5}_{\mathcal{ALCQI}}$ was shown to be decidable and 2-ExpTime-hard. This is the first decidability result for reasoning in temporal schemas with full timestamping—i.e., timestamping for entities, relationships, and attributes.

This paper leaves out few interesting open problems for further investigation. First, a tight complexity bound is not known. The gap between the 2-ExpTime-hardness and the 4-ExpTime algorithm shown in this paper needs to be closed. Second, we believe that the converse translation, from TER with full timestamping to $\mathsf{S5}_{\mathcal{ALCQI}}$, is also possible. Once formally proved, this result would allow us to characterize the complexity of reasoning over TER with timestamping. Finally, we are interested in checking the limits of expressive power of $\mathsf{S5}_{\mathcal{ALCQI}}$ w.r.t. various constraints have appeared in literature on temporal models other than timestamping.

References

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

[2] 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.

[3] 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.

[4] A. Artale, E. Franconi, F. Wolter, and M. Zakharyaschev. A temporal description logic for reasoning about conceptual schemas and queries. In *Proc. of the 8^{th} Joint European Conference on Logics in Artificial Intelligence (JELIA-02)*, volume 2424 of *LNAI*, pages 98–110. Springer, 2002.

[5] A. Artale, C. Lutz, and D. Toman. A description logic of change. Technical report, LTCS-Report 05-06, Technical University Dresden, 2002. see http://lat.inf.tu-dresden.de/research/reports.html.

[6] 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.

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

[8] J. Chomicki and D. Toman. Temporal 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.

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

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

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

[12] 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.

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

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

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

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