ECAI 2012
Luc De Raedt et al. (Eds.)

doi:10.3233/978-1-61499-098-7-390

Complexity of Branching Temporal Description Logics

Víctor Gutiérrez-Basulto and **Jean Christoph Jung** and **Carsten Lutz** [1]

Abstract. We study branching-time temporal description logics (TDLs) based on the DLs $\mathcal{ALC}$ and $\mathcal{EL}$ and the temporal logics CTL and CTL*. The main contributions are algorithms for satisfiability that are more direct than existing approaches, and (mostly) tight elementary complexity bounds that range from PTIME to 2EXPTIME and 3EXPTIME. A careful use of tree automata techniques allows us to obtain transparent and uniform algorithms, avoiding to deal directly with the intricacies of CTL*.

1 Motivation

Classical Description Logics (DLs), such as those that underly the W3C recommendation OWL, are fragments of first order logic and aim at the representation of and reasoning about *static* knowledge. The inability to capture dynamic and temporal aspects has often been criticized because many relevant applications depend on this type of knowledge, for example: (1) in medical ontologies such as SNOMED CT and FMA [9], there are many concepts that can only be accurately described by referring to dynamic aspects; think, for example, of repeating patterns that indicate a disease such as malaria or of findings such as hyperplasia (a proliferation of cells) which potentially develop into a tumor in the future. (2) DLs are used as a language for describing the conceptual model of databases and considerable research has been devoted to extending this approach to capture also the evolution of databases over time [2, 5]. As a reaction to this shortcoming of classical DLs, various kinds of temporal description logics (TDLs) have been proposed, for details please see the surveys [1, 18] and references therein.

A prominent approach to TDLs, originated in [20] and surveyed in [18], is to combine static DLs with temporal logics that are commonly used in hardware and software verification, based on a two-dimensional product-like semantics. While a large body of literature is available for linear-time TDLs based on combinations of DLs with the temporal logic LTL [3, 7, 4, 13], only limited research was devoted to branching-time TDLs based on CTL and CTL* [15, 8]. From the perspective of ontology applications such as those discussed under (1) above, this is slightly surprising because using LTL operators often results in a modeling that is unrealistically strict. As an example, consider the statement 'each student will eventually be a graduate'. In TDLs based on LTL, this is modeled as $\mathsf{Student} \sqsubseteq \Diamond\mathsf{Graduate}$ or $\mathsf{Student} \sqsubseteq \mathsf{Student}\,\mathcal{U}\,\mathsf{Graduate}$, excluding the possibility that a student leaves university without a degree. In TDLs based on CTL, it is possible to use the much more cautious statement $\mathsf{Student} \sqsubseteq \mathbf{E}\mathsf{Student}\,\mathcal{U}\,\mathsf{Graduate}$ based on the existential path quantifier $\mathbf{E}$, stating that there is a *possible future* in which the student obtains a degree and leaving open the possibility of other possible futures. Strict statements such as 'each human will eventually die' can be expressed as $\mathsf{Human} \sqsubseteq \mathbf{A}\Diamond\mathbf{A}\Box\mathsf{Dead}$ based on the universal path quantifier $\mathbf{A}$.

It has been shown in [15] in the context of monodic temporal first-order logic that TDLs based on CTL are typically decidable whereas TDLs based on CTL* have to be appropriately restricted in order to attain decidability: inside concept implications, only state concepts should be allowed, but no path concepts (these correspond to state formulas and path formulas in CTL*). Since decidability is obtained by translating into monadic second order logic on trees, these results only come with a non-elementary upper complexity bound. The aim of this paper is to reconsider branching-time TDLs based on CTL and CTL* (under the mentioned restriction), to develop more direct algorithms for the satisfiability problem, and to analyze the computational complexity. We concentrate on TDLs that are most natural from the perspective of ontology applications: we consider the basic DLs $\mathcal{ALC}$ and $\mathcal{EL}$, allow the application of temporal operators to concepts and (sometimes) to TBox statements (but never to roles), and assume constant domains—please consult [18] for more information on these choices.

Our investigation starts with the TDLs CTL$_{\mathcal{ALC}}$ and CTL$^*_{\mathcal{ALC}}$ in the case where temporal operators can only be applied to concepts (Section 3). We use a uniform approach to both logics that consists of a combination of Pratt-style type elimination and methods based on non-deterministic tree automata. The approach is enabled by the fact that the interaction between the DL dimension and the temporal dimension is limited, similar to the *fusion* of modal logics [14]. Note, however, that fusions correspond to expanding domains while we use constant domains which impose additional technical difficulties. We emphasize that the careful combination of types and existing tree automata for CTL and CTL* allows us to avoid many of the technical intricacies of CTL*, resulting in a rather transparent overall approach. We obtain EXPTIME-completeness for satisfiability in CTL$_{\mathcal{ALC}}$ and 2EXPTIME-completeness for satisfiability in CTL$^*_{\mathcal{ALC}}$, thus the combined logics are computationally no more complex than their components.

As the next step, we stick with CTL$_{\mathcal{ALC}}$ and CTL$^*_{\mathcal{ALC}}$, but additionally allow the application of temporal operators to TBoxes (Section 4). To establish an elementary upper bound, we again use a uniform approach; it is based on a careful combination of alternating 2-way tree automata and non-deterministic tree automata for CTL and CTL*. We obtain a 2EXPTIME upper bound for CTL$_{\mathcal{ALC}}$ and a 3EXPTIME upper bound for CTL$^*_{\mathcal{ALC}}$. For CTL$_{\mathcal{ALC}}$, we prove a matching lower bound using a reduction from the word problem of alternating Turing machines, which shows that, in the presence of temporal TBoxes, the combination of $\mathcal{ALC}$ and CTL results in an increase of computational complexity by one exponential. For CTL$^*_{\mathcal{ALC}}$, the complexity remains open between 2EXPTIME and 3EXPTIME.

[1] Bremen Universität,Germany, email: {victor, jeanjung, clu}@informatik.uni-bremen.de

Finally, we consider the combinations of the inexpressive DL $\mathcal{EL}$ with fragments of CTL, allowing the application of temporal operators to concepts, only (Section 5). The crucial advantage of $\mathcal{EL}$ over $\mathcal{ALC}$ is that it admits efficient (polytime) reasoning and our main aim is to understand in how far this property transfers to a TDL based on $\mathcal{EL}$. It is interesting to note that linear-time TDLs based on $\mathcal{EL}$ and LTL are computationally not very attractive as they turn out to be of the same complexity as the corresponding combination of $\mathcal{ALC}$ and LTL [3]. In the branching time case, we are able to identify a polytime TDL that could be viewed as an analog of non-temporal $\mathcal{EL}$; it includes the temporal operators $\mathbf{E}\Diamond$ and $\mathbf{E}\Box$. Most other versions of $\text{CTL}_{\mathcal{EL}}$ turn out to be hard for PSPACE or EXPTIME.

Proof details are deferred to the appendix of the long version of this paper, made available at http://www.informatik.uni-bremen.de/tdki/research/papers.html.

2 Preliminaries

We introduce $\text{CTL}^*_{\mathcal{ALC}}$ and $\text{CTL}_{\mathcal{ALC}}$. Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be countably infinite sets of *concept names* and *role names*. $\text{CTL}^*_{\mathcal{ALC}}$*-state concepts* C and $\text{CTL}^*_{\mathcal{ALC}}$*-path concepts* $\mathcal{C}$, $\mathcal{D}$ are defined by the grammar

$$\begin{array}{rcl} C & ::= & \top \mid A \mid \neg C \mid C \sqcap D \mid \exists r.C \mid \mathbf{E}\mathcal{C} \\ \mathcal{C},\mathcal{D} & ::= & C \mid \mathcal{C} \sqcap \mathcal{D} \mid \neg\mathcal{C} \mid \bigcirc\mathcal{C} \mid \Box\mathcal{C} \mid \mathcal{C}\mathcal{U}\mathcal{D} \end{array}$$

where A ranges over $\mathsf{N_C}$, r over $\mathsf{N_R}$, C, D over state concepts, and $\mathcal{C}$, $\mathcal{D}$ over path concepts. $\text{CTL}_{\mathcal{ALC}}$ is the fragment of $\text{CTL}^*_{\mathcal{ALC}}$ in which temporal operators $\bigcirc, \Box, \mathcal{U}$ must be immediately preceded by the path quantifier $\mathbf{E}$. Without further qualification, the term *concept* refers to a state concept. As usual, we use $\bot$ to abbreviate the state concept $\neg\top$, $C \sqcup D$ for $\neg(\neg C \sqcap \neg D)$, and $\forall r.C$ for $\neg\exists r.\neg C$; other Boolean operators such as $C \leftrightarrow D$ are defined as usual. In $\text{CTL}^*_{\mathcal{ALC}}$, we additionally use $\mathbf{A}\mathcal{C}$ to abbreviate the state concept $\neg\mathbf{E}\neg\mathcal{C}$ and $\Diamond\mathcal{C}$ for the path concept $\neg\Box\neg\mathcal{C}$. In $\text{CTL}_{\mathcal{ALC}}$, the abbreviations $\mathbf{A}\bigcirc C$, $\mathbf{A}C\mathcal{U}D$, $\mathbf{E}\Diamond C$, and $\mathbf{A}\Diamond C$ are defined as is usual in CTL [11].

A *CTL*$^*_{\mathcal{ALC}}$*-TBox* $\mathcal{T}$ is a finite set of *concept inclusions (CIs)* $C \sqsubseteq D$ with C, D $\text{CTL}^*_{\mathcal{ALC}}$-state concepts. A *CTL*$_{\mathcal{ALC}}$*-TBox* is defined analogously. Note that inclusions between path concepts are not admitted as they result in undecidability [15].

The semantics of classical, non-termporal DLs such as $\mathcal{ALC}$ is given in terms of *interpretations* of the form $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$, where Δ is a non-empty set called the *domain* and $\cdot^{\mathcal{I}}$ is an *interpretation function* that maps each $A \in \mathsf{N_C}$ to a subset $A^{\mathcal{I}} \subseteq \Delta$ and each $r \in \mathsf{N_R}$ to a binary relation $r^{\mathcal{I}} \subseteq \Delta \times \Delta$. The semantics of branching TDLs is given in terms of temporal interpretations, which are infinite trees in which every node is associated with a classical interpretation. For the purposes of this paper, a *tree* is a directed graph $T = (W, E)$ where $W \subseteq (\mathrm{N} \setminus \{0\})^*$ is a prefix-closed non-empty set of *nodes* and $E = \{(w, wc) \mid wc \in W, w \in \mathrm{N}^*, c \in \mathrm{N}\}$ a set of *edges*; we generally assume that $wc \in W$ and $c' < c$ implies $wc' \in W$ and that E is a total relation. The node $\varepsilon \in W$ is the *root* of T. For brevity and since E can be reconstructed from W, we will usually identify T with W.

A *temporal interpretation* is a structure $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w \in W})$ where $T = (W, E)$ is a tree, and for each $w \in W$, $\mathcal{I}_w$ is an interpretation with domain Δ. We usually write $A^{\mathfrak{I},w}$ instead of $A^{\mathcal{I}_w}$, and intuitively $d \in A^{\mathfrak{I},w}$ means that in the interpretation $\mathfrak{I}$, the object d is an instance of the concept name A *at time point* w. Note that each time point shares the same domain Δ, i.e., we make the *constant domain assumption*. Intuitively, this means that objects are not created or destroyed over time; it is the most general choice since expanding, decreasing, and varying domains can all be simulated [14].

We now define the semantics of $\text{CTL}^*_{\mathcal{ALC}}$-concepts. A *path* in a tree $T = (W, E)$ starting at a node w is a minimal set $\pi \subseteq W$ such that $w \in \pi$ and for each $w' \in \pi$, there is a $c \in \mathrm{N}$ with $w'c \in \pi$. We use $\mathsf{Paths}(w)$ to denote the set of all paths starting at the node w. For a path $\pi = w_0 w_1 w_1 \cdots$ and $i \geq 0$, we use $\pi[i]$ to denote w_i and $\pi[i..]$ to denote the path $w_i w_{i+1} \cdots$. The mapping $\cdot^{\mathfrak{I},w}$ is extended from concept names to $\text{CTL}^*_{\mathcal{ALC}}$-state concepts as follows:

$$\begin{array}{rcl} \top^{\mathfrak{I},w} & = & \Delta \\ (C \sqcap D)^{\mathfrak{I},w} & = & C^{\mathfrak{I},w} \cap D^{\mathfrak{I},w} \\ (\exists r.C)^{\mathfrak{I},w} & = & \{d \in \Delta \mid \exists e : (d,e) \in r^{\mathfrak{I},w} \wedge e \in C^{\mathfrak{I},w}\} \\ (\mathbf{E}\,\mathcal{C})^{\mathfrak{I},w} & = & \{d \in \Delta \mid d \in \mathcal{C}^{\mathfrak{I},\pi} \text{ for some } \pi \in \mathsf{Paths}(w)\} \end{array}$$

where $\mathcal{C}^{\mathfrak{I},\pi}$ refers to the extension of $\text{CTL}^*_{\mathcal{ALC}}$-path concepts on a given path π, defined as:

$$\begin{array}{rcl} C^{\mathfrak{I},\pi} & = & C^{\mathfrak{I},\pi[0]} \quad \text{for state concepts } C \\ (\neg\mathcal{C})^{\mathfrak{I},\pi} & = & \Delta \setminus \mathcal{C}^{\mathfrak{I},\pi} \\ (\mathcal{C} \sqcap \mathcal{D})^{\mathfrak{I},\pi} & = & \mathcal{C}^{\mathfrak{I},\pi} \cap \mathcal{D}^{\mathfrak{I},\pi} \\ (\bigcirc\mathcal{C})^{\mathfrak{I},\pi} & = & \{d \in \Delta \mid d \in \mathcal{C}^{\mathfrak{I},\pi[1..]}\} \\ (\Box\mathcal{C})^{\mathfrak{I},\pi} & = & \{d \in \Delta \mid \forall j \geq 0. d \in \mathcal{C}^{\mathfrak{I},\pi[j..]}\} \\ (\mathcal{C}\mathcal{U}\mathcal{D})^{\mathfrak{I},\pi} & = & \{d \in \Delta \mid \exists\, j \geq 0.(d \in \mathcal{D}^{\mathfrak{I},\pi[j..]} \\ & & \qquad \wedge(\forall\, 0 \leq k < j.\, d \in \mathcal{C}^{\mathfrak{I},\pi[k..]}))\}. \end{array}$$

A temporal interpretation $\mathfrak{I}$ is a *model* of a concept C if $C^{\mathfrak{I},\varepsilon} \neq \emptyset$; it is a *model* of a TBox $\mathcal{T}$ if $C^{\mathfrak{I},w} \subseteq D^{\mathfrak{I},w}$ for all $w \in W$ and all $C \sqsubseteq D$ in $\mathcal{T}$. Thus, a TBox $\mathcal{T}$ is interpreted globally in the sense that it has to be satisfied at *every* time point. As an example, consider the TBox

$$\begin{array}{rcl} \mathsf{Student} & \sqsubseteq & \mathbf{E}\Diamond(\mathsf{Graduated} \sqcap \mathbf{A}\Box\exists\mathsf{worksFor.Company}) \\ \mathsf{Prof} & \sqsubseteq & \mathbf{A}(\mathsf{Prof}\,\mathcal{U}\mathsf{Retired} \;\sqcap\; (\mathsf{Retired} \rightarrow \bigcirc\mathsf{Retired})) \end{array}$$

and note that the first CI is formulated in $\text{CTL}_{\mathcal{ALC}}$ while the latter is $\text{CTL}^*_{\mathcal{ALC}}$ proper.

3 $\text{CTL}^*_{\mathcal{ALC}}$ and $\text{CTL}_{\mathcal{ALC}}$: The Basic Case

Our aim is to establish algorithms and tight complexity bounds for deciding satisfiability in $\text{CTL}_{\mathcal{ALC}}$- and $\text{CTL}^*_{\mathcal{ALC}}$, which is the following problem: given a concept C and a TBox $\mathcal{T}$, decide whether there is a model $\mathfrak{I}$ of $\mathcal{T}$ with $C^{\mathfrak{I},\varepsilon} \neq \emptyset$.

Non-deterministic Tree Automata

A crucial ingredient to our approach are nondeterministic Büchi tree automata for CTL and CTL* as described in [17, 22], which we now introduce in some detail. Let Σ be a finite alphabet and $k \geq 1$. A Σ*-labeled* k*-ary tree* is a pair (T, τ) where T is a tree in which every node has exactly k successors and $\tau : T \to \Sigma$ assigns a letter from Σ to each time point. We sometimes identify (T, τ) with τ. A *nondeterministic Büchi tree automaton* (NBTA) over Σ-labeled k-ary trees is a tuple $\mathcal{A} = (Q, \Sigma, Q^0, \delta, F)$ where Q is a finite set of *states*, $Q^0 \subseteq Q$ is the set of *initial states*, $F \subseteq Q$ is a set of *recurring states*, and $\delta : Q \times \Sigma \to 2^{Q^k}$ is the *transition function*.

Let (T, τ) be a Σ-labeled k-ary tree. A *run* of $\mathcal{A}$ on τ is a Q-labeled k-ary tree (T, r) such that $r(\varepsilon) \in Q^0$ and for each node $w \in T$, we have $\langle r(w \cdot 1), \ldots, r(w \cdot k)\rangle \in \delta(r(w), \tau(w))$. The run is *accepting* if for every path $\pi = w_0 w_1 \cdots$ which starts at ε, we have $r(w_i) \in F$ for infinitely many i. The set of trees accepted by

$\mathcal{A}$ is denoted by $L(\mathcal{A})$. The emptiness-problem for NBTAs, which will be used as a part of our algorithm, can be decided in quadratic time [24].

We now assert the existence of NBTAs for CTL and CTL*, as well as their constructability within certain time bounds. We refrain from introducing CTL and CTL* in full detail, and only mention that they are obtained from CTL$_{\mathcal{ALC}}$ and CTL$^*_{\mathcal{ALC}}$ by dropping the constructor $\exists r.C$; their semantics is based on $2^{\mathsf{N_C}}$-labeled trees of unrestricted arity (in this context, we refer to the elements of $\mathsf{N_C}$ as *propositional letters*). Please refer to [11] for full details. We use $\mathsf{pl}(\varphi)$ to denote the set of propositional letters in a CTL*-formula φ. For $n > 0$, we use $\mathsf{Mod}_n(\varphi)$ to denote the set of all $2^{\mathsf{pl}(\varphi)}$-labeled n-ary trees that satisfy φ at the root. Note that φ is satisfiable iff $\mathsf{Mod}_{\#_{\mathbf{E}}(\varphi)} \neq \emptyset$, where $\#_{\mathbf{E}}(\varphi)$ is the number of subformulas of φ that are of the form $\mathbf{E}\psi$.

Theorem 1 ([17, 22]) *For a CTL*-formula φ and $n \geq 0$, one can construct an NBTA $\mathcal{A}_\varphi = (Q, \Sigma, \delta, Q^0, F)$ in time $\mathsf{poly}(|Q| + n)$ such that $L(A_\varphi) = \mathsf{Mod}_n(\varphi)$, $\Sigma = 2^{\mathsf{pl}(\varphi)}$, $|Q| \in 2^{2^{\mathsf{poly}(|\varphi|)}}$, and $|Q| \in 2^{\mathsf{poly}(|\varphi|)}$ when φ is a CTL formula.*

The Decision Procedure

We now describe the uniform decision procedure for satisfiability in CTL$_{\mathcal{ALC}}$ and CTL$^*_{\mathcal{ALC}}$. It yields a tight ExpTime upper bound for the former case and a tight 2ExpTime upper bound for the latter. The lower bounds are inherited from CTL and CTL* [12, 23].

Let C be a concept and $\mathcal{T}$ a TBox, formulated in CTL$^*_{\mathcal{ALC}}$ or its fragment CTL$_{\mathcal{ALC}}$. We assume w.l.o.g. that $\mathcal{T}$ is of the form $\{\top \sqsubseteq C_\mathcal{T}\}$ and use $\mathsf{cl}(\mathcal{T})$ to denote the set of state concepts that occur in $\mathcal{T}$, closed under subconcepts and single negation. A *type* for $\mathcal{T}$ is a set $t \subseteq \mathsf{cl}(\mathcal{T})$ such that $C_\mathcal{T} \in t$. A *temporal type* for $\mathcal{T}$ has the form (t, i) with t a type for $\mathcal{T}$ and $i \geq 0$ a *distance* that denotes how far a time point w of a tree structure is from the root (i.e., the length $|w|$ of the word w). For any $n \geq 0$, we use $\mathsf{ttp}_n(\mathcal{T})$ to denote the set of all temporal types (t, i) for $\mathcal{T}$ with $i \leq n$. The algorithm starts with the set of temporal types $\mathsf{ttp}_{n_0}(\mathcal{T})$ for some appropriate bound n_0 to be determined later and then generates a decreasing sequence $S_0 \supseteq S_1 \supseteq \ldots$ where $S_0 = \mathsf{ttp}_{n_0}(\mathcal{T})$ and S_{j+1} is obtained from S_j by eliminating temporal types that, intuitively, cannot occur in any model of $\mathcal{T}$. The algorithm terminates when no further types are eliminated, i.e., when $S_j = S_{j+1}$. It returns "satisfiable" if there is a surviving (t, i) with $C \in t$ and $i = 0$, and "unsatisfiable" otherwise.

We now formally describe the elimination condition. For a type t, let $\bar{t}$ denote the result of replacing every concept $C \in t \setminus \mathsf{N_C}$ with a fresh concept name X_C, and let cn denote the set of all resulting concept names, including those in $\mathcal{T}$. For $C \in \mathsf{cl}(\mathcal{T})$, let $\overline{C}$ denote the result of replacing in C every subconcept $\exists r.D$ with $X_{\exists r.D}$. Let $\sharp_{\mathbf{E}}(\mathcal{T})$ denote the number of state concepts in $\mathsf{cl}(\mathcal{T})$ that are of the form $\mathbf{E}\mathcal{C}$. A temporal type (t, i) is removed from S_j if it violates one of the following:

1. if $\exists r.C \in t$, then there is a $(t', i) \in S_j$ such that $\{C\} \cup \{\neg D \mid \neg\exists r.D \in t\} \subseteq t'$;
2. (t, i) is *S_j-realizable*, i.e., there is a 2^{cn}-labeled $\sharp_{\mathbf{E}}(\mathcal{T})$-ary tree (T, τ) that satisfies the following conditions, where $\rho(i) = \min\{n_0, i\}$:
   (a) for some $w \in T$ with $|w| = i$, we have $\tau(w) = \bar{t}$;
   (b) for each $w \in T$, there is a $(t, \rho(i)) \in S_j$ with $\tau(w) = \bar{t}$;
   (c) ε satisfies $\mathbf{A}\Box \bigwedge_{X_C \in \mathsf{cn}} X_C \leftrightarrow \overline{C}$.

Condition 1 takes care of the DL dimension of CTL$^*_{\mathcal{ALC}}$ while Condition 2 takes care of the (Boolean constructors and the) temporal dimension; intuitively, the tree (T, τ) describes the temporal evolution of a single domain element. The intuition behind the number n_0 and the use of $\rho(\cdot)$ in Condition 2 is that time points which are close to the root of the structure behave in a special way. For example, when $\mathcal{T} = \{\top \sqsubseteq \mathbf{A}\bigcirc\bigcirc\neg A\}$, then time points w with distance $|w| < 2$ are special in that they can satisfy A. Using binary counting, one can construct similar examples where time points with exponential distance are still special; see [18] for a similar observation for LTL$_{\mathcal{ALC}}$. The final result S of type elimination represents the infinite expansion $S_\omega := S \cup \{(t, m) \mid (t, n_0) \in S \wedge m > n_0\}$. For being able to build a model, we want all $(t, i) \in S_\omega$ to satisfy Conditions 1 and 2 when, in Condition 2, $\rho(i)$ is replaced with i. This suggests the main property to attain by choosing an appropriate bound n_0:

($*$) if $(t, n_0) \in S$ is S-realizable, then $(t, n_0 + \ell)$ is S-realizable for any $\ell \geq 0$.

One might be tempted to choose $n_0 = |\mathsf{tp}(\mathcal{T})|$. While this is indeed sufficient for CTL$_{\mathcal{ALC}}$, it does not work for CTL$^*_{\mathcal{ALC}}$, where types do not capture enough information about models and time points of double exponential distance can still be special. To solve this problem, we observe that NBTAs can used to verify Condition 2 above, and that this suggests a concrete bound n_0. Specifically, let φ be the formula from Condition 2(c) and $\mathcal{A}_\varphi$ the corresponding NBTA from Theorem 1 with set of states Q.

Lemma 1 *When choosing $n_0 := |Q| \cdot |\mathsf{tp}(\mathcal{T})|$ as a bound for the type elimination procedure, then Property ($*$) is satisfied and "satisfiable" is returned iff C is satisfiable w.r.t. $\mathcal{T}$.*

The proof of the first part of Lemma 1 that asserts satisfaction of ($*$) is rather subtle and involves a very careful use of automata techniques. We have not yet said how NBTAs can be used to verify Condition 2. The idea is to construct three NBTAs, one for each of the parts (a) to (c), build the intersection NBTA which accepts precisely the 2^{cn}-trees required for Condition 2, and then to perform an emptiness test. For part (c), we can simply use $\mathcal{A}_\varphi$. Moreover, it is easy to define an NBTA $\mathcal{A}_{t,i}$ with $i \leq n_0$ states that verifies the condition in part (a), and the same is true for part (b) and an NBTA $\mathcal{A}_{S_j}$ with n_0 states. Details are left to the reader.

It remains to show that the algorithm runs in double exponential time in the case of CTL$^*_{\mathcal{ALC}}$ and in exponential time for CTL$_{\mathcal{ALC}}$. We use $|\mathcal{T}|$ to denote the *size* of $\mathcal{T}$, which is the number of symbols needed to write it. The bound n_0 is in $O(2^{2^{\mathsf{poly}(|\mathcal{T}|)}})$ for CTL$_{\mathcal{ALC}}$ and in $O(2^{\mathsf{poly}(|\mathcal{T}|)})$ for CTL$^*_{\mathcal{ALC}}$. The number of steps of the type elimination procedure is bounded by $2^{O(|\mathcal{T}|)} \cdot n_0$. The number of states in $\mathcal{A}_\varphi$ is n_0 and thus it remains to recall that the intersection of a constant number of NBTAs can be constructed with only a polynomial blowup and that emptiness can be decided in quadratic time.

Theorem 2 *Satisfiability is* ExpTime*-complete for CTL$_{\mathcal{ALC}}$ and* 2ExpTime*-complete for CTL$^*_{\mathcal{ALC}}$.*

4 CTL$^*_{\mathcal{ALC}}$ and CTL$_{\mathcal{ALC}}$: Temporal TBoxes

We again study satisfiability of CTL$^*_{\mathcal{ALC}}$- and CTL$_{\mathcal{ALC}}$-TBoxes, but now allow temporal operators to be applied also to concept inclusions in a TBox. *CTL$^*_{\mathcal{ALC}}$-state TBoxes* φ and *CTL$^*_{\mathcal{ALC}}$-path TBoxes* ψ, ϑ are formed according to the grammar

$$\begin{aligned} \varphi &::= C \sqsubseteq D \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{E}\psi \\ \psi, \vartheta &::= \varphi \mid \neg\psi \mid \vartheta \wedge \psi \mid \bigcirc\psi \mid \psi \mathcal{U} \vartheta. \end{aligned}$$

We define truth relations $\mathfrak{I}, w \models \varphi$ and $\mathfrak{I}, \pi \models \psi$ (where $\mathfrak{I}$ is a temporal model, w a time point in $\mathfrak{I}$, and π a path in $\mathfrak{I}$) in the obvious way, c.f. Section 2; in particular, $\mathfrak{I}, w \models C \sqsubseteq D$ iff $C^{\mathfrak{I},w} \subseteq D^{\mathfrak{I},w}$. A *temporal CTL$^*_{\mathcal{ALC}}$-TBox* is a CTL$^*_{\mathcal{ALC}}$-state TBox; temporal CTL$_{\mathcal{ALC}}$-TBoxes are defined in the expected way. We say that $\mathfrak{I}$ is a *model* of a temporal CTL$^*_{\mathcal{ALC}}$-TBox φ if $\mathfrak{I}, \varepsilon \models \varphi$. Temporal TBoxes are useful for expressing the dynamics of policies; for example, the temporal CTL$_{\mathcal{ALC}}$-TBox

$$\mathbf{A}\Diamond(\mathsf{Student} \sqcap \exists \mathsf{fails}.\mathsf{MajorExam} \sqsubseteq \mathbf{A}\Box\neg\mathsf{Student})$$

says that, in all possible futures, there will be a policy such that all students who fail a single major exam will immediately and lastingly be exmatriculated.

Alternating Automata

To derive algorithms and upper bounds for the satisfiability of temporal TBoxes, we use a careful mixture of NBTAs and *alternating* Büchi tree automata. More precisely, an *alternating 2-way Büchi tree automaton (2ABTA)* over Σ-labeled k-ary trees is a tuple $\mathcal{A} = (Q, \Sigma, Q^0, \delta, F)$ where all components except δ are as for NBTAs. For a set X, let $\mathcal{B}^+(X)$ be the set of Boolean formulas built from elements in X using $\wedge$, $\vee$, true and false. Let $Y \subseteq X$. We say that Y *satisfies* a formula $\theta \in \mathcal{B}^+(X)$ if assigning true to the members of Y and assigning false to the members of $X \setminus Y$ makes θ true. Let $[k] = \{-1, 0, \ldots, k\}$. For any $w \in (\mathrm{N} \setminus \{0\})^*$ and $m \in k$, we put $\mathsf{mov}(w, m) = w$ if $m = 0$, $\mathsf{mov}(w, m) = w \cdot m$ if $m > 0$, and $\mathsf{mov}(w, m) = u$ if $m = -1$ and $w = uc$ with $c \in \mathrm{N}$. The transition function δ of a 2ABTA is a function $\delta : Q \times \Sigma \times \{\mathsf{t}, \mathsf{f}\} \to \mathcal{B}^+([k] \times Q)$.

Let (T, τ) be a Σ-labeled k-ary tree. For $w \in T$, put $\mathsf{root}(w) = \mathsf{t}$ if $w = \varepsilon$ and $\mathsf{root}(w) = \mathsf{f}$ otherwise. A *run* of $\mathcal{A}$ on τ is a $T \times Q$-labeled tree (T_r, r) such that $r(\varepsilon) = (\varepsilon, q_0)$ for some $q_0 \in Q^0$ and whenever $x \in T_r, r(x) = (w, q)$, and $\delta(q, \tau(w), \mathsf{root}(w)) = \theta$, then there is a set $\mathcal{S} = \{(m_1, q_1), \ldots, (m_n, q_n)\} \subseteq [k] \times Q$ such that $\mathcal{S}$ satisfies θ and for $1 \leq i \leq n$, we have $x \cdot i \in T_r$, $\mathsf{mov}(w, m_i)$ is defined, and $\tau_r(x \cdot i) = (\mathsf{mov}(w, m_i), q_i)$. The emptiness problem for 2ABTAs is EXPTIME-complete [21]. Using the root flag as an additional third component in the transition function is slightly unorthodox, but easily seen to not cause any problems. It will allow use to construct more compact 2ABTAs later on.

The Decision Procedure

Let φ be a temporal CTL$^*_{\mathcal{ALC}}$-TBox whose satisfibility is to be decided. We use $\mathsf{cl}(\varphi)$ to denote the set of state concepts that occur in φ, closed under subconcepts and single negation. A *concept type* for φ is a set $t \subseteq \mathsf{cl}(\varphi)$ and $\mathsf{tp}(\varphi)$ denotes the set of all concept types for $\mathcal{T}$. We use $\mathsf{sub}(\varphi)$ to denote the set of all state subformulas of φ.

A *quasi-world for* φ is a pair (S_1, S_2) with $S_1 \subseteq \mathsf{tp}(\varphi)$ a set of concept types and $S_2 \subseteq \mathsf{sub}(\varphi)$ a formula type such that

1. if $t \in S_1$ and $\exists r.C \in t$, then there is a $t' \in S_1$ with $\{C\} \cup \{\neg D \mid \neg\exists r.D \in t\} \subseteq t'$;
2. for all $C \sqsubseteq D \in \mathsf{sub}(\varphi)$, we have $C \sqsubseteq D \in S_2$ iff, for all $t \in S_1$, $C \in t$ implies $D \in t$.

Let $\mathsf{qw}(\varphi)$ denote the set of all quasi-worlds for φ. A *quasi-model* $\mathfrak{M}$ *for* φ is a $\mathsf{qw}(\varphi)$-labeled tree, of any outdegree.

For $t \in \mathsf{tp}(\varphi)$, $\overline{t}$ is the result of replacing every $C \in t \setminus \mathsf{N_C}$ with a fresh concept name X_C, and cn_X denotes the set of all resulting concept names, including those in $\mathcal{T}$. For $C \in \mathsf{cl}(\mathcal{T})$, $\overline{C}$ denotes the result of replacing in C every subconcept $\exists r.D$ with $X_{\exists r.D}$. For every $\psi \in \mathsf{sub}(\varphi)$, $\overline{\psi}$ denotes the result of replacing every subformula $C \sqsubseteq D$ of ψ with a fresh concept name Y_ψ (which plays the role of a propositional letter for CTL / CTL*) and cn_Y is the set of all concept names thus introduced. For $S \subseteq \mathsf{sub}(\varphi)$, we set $\overline{S} = \{\overline{\psi} \mid \psi \in S\}$. For $\mathfrak{M}$ a quasi-model, we use $\mathfrak{M}_2$ to denote the 2^{cn_Y}-labeled tree obtained by associating each node $w \in \mathfrak{M}$ with the label $\overline{S_2(w)}$.

A quasi-model $\mathfrak{M} = (T, \tau)$ is *proper* if the following conditions are satisfied:

1. $\mathfrak{M}_2 \models \overline{\varphi}$;
2. for all $w \in T$ with $\tau(w) = (S_1, S_2)$ and all $s \in S_1$, there is a 2^{cn_X}-labeled tree (T, τ') such that
   (a) $\tau'(w) = \overline{s}$;
   (b) for all $w' \in T$ with $\tau(w') = (S_1', S_2')$, there is an $s' \in S_1'$ such that $\tau'(w') = \overline{s'}$;
   (c) ε satisfies $\mathbf{A}\Box \bigwedge_{X_C \in \mathsf{cn}_X} (X_C \leftrightarrow \overline{C})$.

Intuitively, Condition 1 ensures that $\mathfrak{M}$ satisfies the temporal TBox φ and Condition 2 guarantees that, for each required domain element, we can consistently select a type from the quasi-world at each node of $\mathfrak{M}$. The following result shows that to decide satisfiability of φ, it suffices to check the existence of a proper quasi-model for φ.

Proposition 2 *φ is satisfiable iff there is a proper quasi-model for φ.*

The following NBTAs will be used in our decision procedure. Let ϑ be the formula in Condition 2(c). By Theorem 1, we find an NBTA $\mathcal{A}_{\overline{\varphi}} = (Q_1, \Sigma_1, \delta_1, Q_1^0, F_1)$ that accepts exactly the 2^{cn_Y}-labeled $\#^f_{\mathbf{E}}(\varphi)$-ary trees which satisfy $\overline{\varphi}$, where $\#^f_{\mathbf{E}}(\varphi)$ denotes the set of state formulas of the form $\mathbf{E}\psi$ in $\mathsf{sub}(\varphi)$; we also find an NBTA $\mathcal{A}_\vartheta = (Q_2, \Sigma_2, \delta_2, Q_2^0, F_2)$ that accepts exactly the 2^{cn_X}-labeled $\#^c_{\mathbf{E}}(\varphi)$-ary trees which satisfy ϑ, where $\#^c_{\mathbf{E}}(\varphi)$ denotes the set of state concepts of the form $\mathbf{E}\mathcal{C}$ in $\mathsf{sub}(\varphi)$.

We aim at constructing a 2ABTA $\mathcal{A}$ on $\mathsf{qw}(\varphi)$-labeled trees that accepts precisely the proper quasi-models for φ. For doing this, we have to restrict the outdegree of quasi-models in an appropriate way. Set $k := |\mathsf{qw}(\varphi)| \cdot |\mathsf{tp}(\varphi)| \cdot |Q_2|$. The following is proved by replacing Condition 2(c) with a version based on the NBTA $\mathcal{A}_\vartheta$ and carefully analyzing its runs.

Lemma 3 *There is a proper quasi-model for φ iff there is a quasi-model for φ that is a k-ary tree.*

The desired 2ABTA $\mathcal{A}$ will thus run on k-ary trees. For simplicity and because Theorem 1 admits any outdegree, we can actually assume both $\mathcal{A}_{\overline{\varphi}}$ and $\mathcal{A}_\vartheta$ to run on trees of outdegree k (this does not result in a change to the state set Q_2, thus does not impact k). Since 2ABTAs are trivially closed under intersection, it suffices to construct separate 2ABTAs $\mathcal{A}_1$ and $\mathcal{A}_2$ to deal with Conditions 1 and 2 of proper quasi-models. To obtain $\mathcal{A}_1$, manipulate $\mathcal{A}_{\overline{\varphi}}$ so that it has input alphabet $\mathsf{qw}(\varphi)$ and each symbol (S_1, S_2) is treated as $\overline{S_2}$, and view the resulting automaton as a 2ABTA. The 2ABTA $\mathcal{A}_2 = (Q, \Sigma, \delta, \{q_0\}, F)$ verifies Condition 2 by simulating a run of $\mathcal{A}_\vartheta$ for every $w \in T$ with $\tau(w) = (S_1, S_2)$ and every $s \in S_1$. Formally, set $Q_2^* = Q_2 \cup \{*\}$ and

$$Q = \{q_0\} \cup (Q_2 \times Q_2^*) \cup (Q_2 \times 2^{\mathsf{cn}_X} \times Q_2^*)$$

and the transition relation δ is as follows, for $\omega = (S_1, S_2)$:

$$\delta(q_0, \omega, \cdot) = \bigwedge_{i=1}^{k}(i, q_0) \wedge \bigwedge_{s\in S_1} \bigvee_{q\in Q_2}(0, (q, s, *))$$

$$\delta((q, q'), \omega, \cdot) = \bigvee_{s\in S_1}(0, (q, s, q'))$$

$$\delta((q, s, q'), \omega, \mathtt{t}) = \bigvee_{(q_1,\ldots,q_k)\in\delta_2(q,s)\mid q'\in\{q_1,\ldots,q_k\}} \bigwedge_{i=1}^{k}(i, (q_i, *))$$

$$\delta((q, s, q'), \omega, \mathtt{f}) = \bigvee_{p\in Q_2}(-1, (p, q'))) \wedge \bigvee_{(q_1,\ldots,q_k)\in\delta_2(q,s)\mid q'\in\{q_1,\ldots,q_k\}} \bigwedge_{i=1}^{k}(i, (q_i, *))$$

where $\cdot$ in the third component means that the transition exists both when the component is $\mathtt{t}$ and $\mathtt{f}$, and '$*$' behaves like a wildcard for all states of Q_2 with the test $* \in \{q_1, \ldots, q_k\}$ always being successfull. Finally, we set $F = F_2$. Note that runs of the original NBTA $\mathcal{A}_\vartheta$ must start at the root of the tree, but when simulating $\mathcal{A}_\vartheta$ in $\mathcal{A}$, we have to start at an arbitrary tree node. In fact, this is the reason why we need a *2-way* automaton and states of the form (q, q') and (q, s, q'), which intuitively mean that we are currently simulating a run of $\mathcal{A}_\vartheta$ in state q and have already decided to assign q' to some successor of the current node (we do not need to memorize *which* successor since the transitions of $\mathcal{A}_\vartheta$ are closed under permuting the successors). The state (q, s, q') additionally selects an $s \in S_1$ for the current tree node, see Condition 2. A careful analysis shows that our approach yields the following upper bounds.

Theorem 3 *Satisfiability of temporal TBoxes is in* 2ExpTime *for CTL$_{\mathcal{ALC}}$ and in* 3ExpTime *for CTL$^*_{\mathcal{ALC}}$.*

For CTL$_{\mathcal{ALC}}$, we can establish a matching 2ExpTime lower bound by reducing the word problem of exponentially space-bounded, alternating Turing machine (ATM). The reduction is too long to be presented here in full detail, so we only sketch some central ideas. Assume an ATM $\mathcal{M}$ and an input word α to $\mathcal{M}$ are given. We construct a temporal CTL$_{\mathcal{ALC}}$-TBox $\varphi_{\mathcal{M},\alpha}$ such that models of $\varphi_{\mathcal{M},\alpha}$ correspond to accepting computation trees of $\mathcal{M}$ on α. In particular, the computation tree is represented by the temporal development of a single domain element d_0 with each time point w corresponding to a tape cell and a configuration of $\mathcal{M}$ being represented by exponentially many consecutive time points. A major challenge is to transport information (a symbol found on a type cell) exponentially far down the tree using a polysize TBox. The solution is to store the information in additional domain elements generated with existential restrictions; to recover the stored information to the 'main' domain element d_0, we cannot use r since roles can vary freely over time; instead, we use the temporal TBox to exchange information between domain elements. In a nutshell, this can be done by temporal TBox statements such as

$$\mathbf{A}\Box(\top \sqsubseteq A \vee \top \sqsubseteq \neg A)$$

which ensures that the truth value of A, and thus a single bit of information, is shared by all domain elements. To transport symbols in our ATM reduction, we need to refine this basic idea, for example by using a suitable set of binary counters to manage distances in the tree. The resulting TBox $\varphi_{\mathcal{M},\alpha}$ has the form $\mathbf{A}\Box\psi$ with ψ a Boolean combination of CIs $C \sqsubseteq D$.

Theorem 4 *Satisfiability of temporal CTL$_{\mathcal{ALC}}$-TBoxes is* 2ExpTime*-complete.*

5 Fragments of CTL$_{\mathcal{EL}}$

The $\mathcal{EL}$-family of DLs is a popular family of lightweight ontology languages [6] whose key feature is to admit polytime reasoning while still providing sufficient expressiveness for many applications. In particular, members of the $\mathcal{EL}$-family are used in medical ontologies such as SNOMED CT and underlie the OWL 2 EL profile of the OWL 2 ontology language. We consider fragments of CTL$_{\mathcal{EL}}$, the fragment of CTL$_{\mathcal{ALC}}$ that disallows the constructor $\neg$ (and thus also the abbreviations $C \sqcup D$ and $\forall r.C$). Throughout this section, we only allow the application of temporal operators to concepts, but not to TBoxes. As an example, consider the following CTL$_{\mathcal{EL}}$-TBox:

$$\begin{aligned} \mathsf{PhDStudent} &\sqsubseteq \mathbf{E}\Diamond(\mathsf{Phd} \sqcap \mathbf{E}\Diamond\exists\mathsf{worksFor.Uni}),\\ \exists\mathsf{worksFor.Uni} &\sqsubseteq \mathbf{E}\Diamond\mathbf{E}\Box\mathsf{Professor} \end{aligned}$$

Because of the absence of negation, satisfiability in CTL$_{\mathcal{EL}}$ is trivial; as in non-temporal $\mathcal{EL}$, we therefore consider subsumption as the central reasoning problem. Formally, a concept D *subsumes* a concept C w.r.t. a TBox $\mathcal{T}$, written $\mathcal{T} \models C \sqsubseteq D$, if $C^{\mathfrak{I}} \subseteq D^{\mathfrak{I}}$ for all temporal interpretations $\mathfrak{I}$ that are a model of $\mathcal{T}$. For example, the above TBox implies that every PhD student has the possible future of becoming a professor, formally $\mathcal{T} \models \mathsf{PhDStudent} \sqsubseteq \mathbf{E}\Diamond\mathsf{Professor}$.

With the aim of identifying a computationally efficient branching-time TDL, we consider various fragments of CTL$_{\mathcal{EL}}$ obtained by admitting sets of temporal operators. In this context, we view each operator from the set $\mathbf{E}\bigcirc C$, $\mathbf{A}\bigcirc C$, $\mathbf{E}\Diamond C$, $\mathbf{E}\Box C$, $\mathbf{A}\Diamond C$, $\mathbf{A}\Box C$, $\mathbf{E}C\mathcal{U}D$ and $\mathbf{A}C\mathcal{U}D$ as primitive instead of as an abbreviation. For uniformity, we denote fragments of CTL$_{\mathcal{EL}}$ by putting the available temporal operators in superscript; for example, CTL$_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\Box}$ is CTL$_{\mathcal{EL}}$ with only the operators $\mathbf{E}\Diamond$ and $\mathbf{E}\Box$. We obtain a landscape of temporal variants of $\mathcal{EL}$ with the complexity of subsumption ranging from PTime over PSpace-hard to ExpTime-complete.

A tractable fragment

We assume that the input TBox is in the following normal form. A *basic concept* is a concept of the form $\top, A, \exists r.A, \mathbf{E}\Diamond A, \mathbf{E}\Box A$ where A is a concept *name*. Now, every CI in the input TBox is required to be of the form

$$X_1 \sqcap \ldots \sqcap X_n \sqsubseteq X$$

with $X_1, \ldots, X_n, X$ basic concepts. Every TBox can be transformed into this normal form in polytime such that (non-)subsumption between the concept names that occur in the original TBox is preserved, c.f. [6]. We show that concept subsumption w.r.t. CTL$_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\Box}$-TBoxes can be decided in polynomial time by reducing it to subsumption in the extension $\mathcal{EL}^{++}$ [6] of $\mathcal{EL}$. In particular, $\mathcal{EL}^{++}$ allows to specify properties on roles, such as reflexivity, transitivity, and role hierarchy statements of the form $r \sqsubseteq s$. We introduce fresh role names $\mathsf{succ}_\Diamond$ and $\mathsf{succ}_\Box$. Intuitively, a role name $\mathsf{succ}_\Diamond$ represents the 'going on step to the future' relation and a subrole $\mathsf{succ}_\Box$ of $\mathsf{succ}_\Diamond$ is used to deal with concepts of the form $\mathbf{E}\Box A$. We require that

- $\mathsf{succ}_\Diamond$ is *transitive* and *reflexive*,
- $\mathsf{succ}_\Diamond$ and $\mathsf{succ}_\Box$ are *total*; and
- $\mathsf{succ}_\Box \sqsubseteq \mathsf{succ}_\Diamond$.

We obtain an $\mathcal{EL}^{++}$-TBox $\mathcal{T}'$ from a CTL$_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\Box}$-TBox $\mathcal{T}$ by (i) replacing every subconcept $\mathbf{E}\Diamond A$ with $\exists\mathsf{succ}_\Diamond.A$, (ii) replacing every subconcept $\mathbf{E}\Box A$ with M_A for some fresh concept name M_A, (iii) adding for each fresh concept name M_A introduced in step (ii) the concept inclusion

$$M_A \sqsubseteq A \sqcap \exists\mathsf{succ}_\Box.M_A$$

and (iv) including the properties of roles listed above. Note that the role inclusion $\mathsf{succ}_\Box \sqsubseteq \mathsf{succ}_\Diamond$ is needed since $\emptyset \models \mathbf{E}\Box A \sqsubseteq \mathbf{E}\Diamond A$. It is now possible to show the following.

Lemma 4 *Let A, B be two concept names occurring in $\mathcal{T}$. Then, $\mathcal{T} \models A \sqsubseteq B$ iff $\mathcal{T}' \models A \sqsubseteq B$.*

Since concept subsumption in $\mathcal{EL}^{++}$ can be decided in PTIME [6], we obtain the desired result.

Theorem 5 *In $CTL_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\Box}$, subsumption can be decided in* PTIME.

We note that this is the first temporal description logic based on $\mathcal{EL}$ that turns out to admit PTIME reasoning; see also [3]. While the expressive power of $\text{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\Box}$ is clearly rather restricted, we believe that it might still be sufficient for some applications. In some sense, the situation parallels the one for non-temporal $\mathcal{EL}$. Note that the example given at the beginning of this section is formulated in $\text{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\Box}$.

Intractable Fragments

We show that $\text{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\Box}$ is a maximal tractable fragment of $\text{CTL}_{\mathcal{ALC}}$ in the sense that adding further temporal operators destroys tractability. We start with the extension $\text{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\Box,\mathbf{A}\Box}$ and prove the following by a reduction from QBF validity. Since the strategy of the reduction is rather standard, we defer details to the technical report.

Theorem 6 *Subsumption in $CTL_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\Box,\mathbf{A}\Box}$ is* PSPACE*-hard.*

We conjecture that $\mathcal{EL}^{\mathbf{E}\Diamond,\mathbf{E}\Box,\mathbf{A}\Box}$ is actually PSPACE-complete, but leave an upper bound as future work.

The remaining candidate operators for extending $\text{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\Box}$ are $\mathbf{E}\bigcirc$, $\mathbf{A}\bigcirc$, $\mathbf{A}\Diamond$, $\mathbf{E}\mathcal{U}$, $\mathbf{A}\mathcal{U}$. It turns out that subsumption is EXPTIME-complete in any of the resulting extensions. In fact, one does not even need both temporal operators from $\text{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\Box}$ for the lower bounds.

Theorem 7 *Subsumption is* EXPTIME*-complete in*

(a) $CTL_{\mathcal{EL}}^{\mathbf{A}\Diamond,\mathbf{E}\Diamond}$ *(b)* $CTL_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\bigcirc}$ *(c)* $CTL_{\mathcal{EL}}^{\mathbf{A}\Diamond,\mathbf{A}\bigcirc}$
(d) $CTL_{\mathcal{EL}}^{\mathbf{E}\mathcal{U}}$ *(e)* $CTL_{\mathcal{EL}}^{\mathbf{A}\mathcal{U}}$ *(f)* $CTL_{\mathcal{EL}}^{\mathbf{A}\bigcirc}$

The upper bounds are obvious since all listed TDLs are a fragment of $\text{CTL}_{\mathcal{ALC}}$, and the lower bounds are established as follows. It is well-known that every *non-convex* extension of $\mathcal{EL}$ is at least as hard as $\mathcal{ALC}$, where convexity means that whenever $\mathcal{T} \models C \sqsubseteq D_1 \sqcup \cdots \sqcup D_n$ with $n \geq 2$, then $\mathcal{T} \models C \sqsubseteq D_i$ for some i [6]. The same is true for non-convex fragments of $\text{CTL}_{\mathcal{EL}}$ and $\text{CTL}_{\mathcal{ALC}}$. To establish the lower bound in Theorem 7, it thus suffices to argue that the listed fragments are not convex. For example, consider $\text{CTL}_{\mathcal{EL}}^{\mathbf{A}\Diamond,\mathbf{E}\Diamond}$, set $\mathcal{T} = \emptyset$ and

$$\begin{aligned} C &= \mathbf{A}\Diamond A \sqcap \mathbf{A}\Diamond B \\ D_1 &= \mathbf{E}\Diamond(A \sqcap \mathbf{E}\Diamond B) \\ D_2 &= \mathbf{E}\Diamond(B \sqcap \mathbf{E}\Diamond A) \end{aligned}$$

Clearly, $\mathcal{T} \models C \sqsubseteq D_1 \sqcup D_2$, but neither $\mathcal{T} \models C \sqsubseteq D_1$ nor $\mathcal{T} \models C \sqsubseteq D_2$. Most remaining cases are similar to related fragments of $\text{LTL}_{\mathcal{EL}}$ studied in [3] and are treated in detail in the technical report.

The logic $\text{CTL}_{\mathcal{EL}}^{\mathbf{A}\bigcirc}$ is an exceptional case since it can be proved to be convex. However, it is nevertheless EXPTIME-hard, which follows from the observation that, after dropping the contructor $\exists r.C$, $\text{CTL}_{\mathcal{EL}}^{\mathbf{A}\bigcirc}$ is a notational variant of the description logic $\mathcal{FL}_0$ which is shown to be EXPTIME-complete in [6, 16].

6 Conclusion

As future work, it would be interesting to determine the precise complexity of satisfiability of temporal $\text{CTL}^*_{\mathcal{ALC}}$-TBoxes, which is currently open between 2EXPTIME and 3EXPTIME, and to analyze branching-time TDLs based on the DL-Lite family of DLs. It also seems natural to generalize the expressive power of the branching time component as demanded by applications. This includes capturing statements such as '*it is likely* that an irregular mole develops into a melanoma in the future' and 'all students will graduate within 8 semesters'.

REFERENCES

[1] Alessandro Artale and Enrico Franconi, 'Temporal description logics', in *Handbook of Time and Temporal Reasoning in Artificial Intelligence*, pp. 375–388, 2005.

[2] Alessandro Artale, 'Reasoning on temporal conceptual schemas with dynamic constraints', in *TIME*, pp. 79–86, 2004.

[3] Alessandro Artale, Roman Kontchakov, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev, 'Temporalising tractable description logics', in *TIME*, pp. 11–22. IEEE Computer Society, (2007).

[4] Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev, 'Past and future of DL-Lite', in *AAAI*, 2010.

[5] Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev, 'Tailoring temporal description logics for reasoning over temporal conceptual models', in *FroCos*, volume 6989 of *LNCS*, pp. 1–11, 2011.

[6] Franz Baader, Sebastian Brandt, and Carsten Lutz, 'Pushing the $\mathcal{EL}$ envelope', in *IJCAI*, pp. 364–369, 2005.

[7] Franz Baader, Silvio Ghilardi, and Carsten Lutz, 'LTL over description logic axioms', in *KR*, pp. 684–694, 2008.

[8] Sebastian Bauer, Ian M. Hodkinson, Frank Wolter, and Michael Zakharyaschev, 'On non-local propositional and weak monodic quantified CTL', *J. Log. Comput.*, **14**(1), pp. 3–22, 2004.

[9] Olivier Bodenreider and Songmao Zhang, 'Comparing the representation of anatomy in the FMA and SNOMED CT', in *AMIA Annual Symposium*, pp. 46–50, 2006.

[10] Ashok K. Chandra, Dexter Kozen, and Larry J. Stockmeyer, 'Alternation', *J. ACM*, **28**(1), pp. 114–133, 1981.

[11] Edmund. M. Clarke, Orna Grumberg, and Doron Peled, *Model checking*, MIT Press, 1999.

[12] Michael J. Fischer and Richard E. Ladner, 'Propositional dynamic logic of regular programs', *J. Comput. Syst. Sci.*, **18**(2), pp. 194–211, 1979.

[13] Enrico Franconi and David Toman, 'Fixpoints in temporal description logics', in *IJCAI*, pp. 875–880, 2011.

[14] Dov Gabbay, Agi Kurucz, Frank Wolter, and Michael Zakharyaschev, *Many-dimensional modal logics: theory and applications*, Studies in Logic, 148, 2003.

[15] Ian M. Hodkinson, Frank Wolter, and Michael Zakharyaschev, 'Decidable and undecidable fragments of first-order branching temporal logics', in *LICS*, pp. 393–402. 2002.

[16] Martin Hofmann, 'Proof-theoretic approach to description-logic', in *LICS*, pp. 229–237, 2005.

[17] Orna Kupferman and Moshe Y. Vardi, 'Safraless decision procedures', in *FOCS*, pp. 531–542, 2005.

[18] Carsten Lutz, Frank Wolter, and Michael Zakharyaschev, 'Temporal description logics: A survey', in *TIME*, pp. 3–14, 2008.

[19] Christos H. Papadimitriou, *Computational complexity*, Academic Internet Publ., 2007.

[20] Klaus Schild, 'Combining terminological logics with tense logic', in *EPIA*, volume 727 of *LNCS*, pp. 105–120, (1993).

[21] Moshe Y. Vardi, 'Reasoning about the past with two-way automata', in *ICALP*, volume 1443 of *LNCS*, pp. 628–641, 1998.

[22] Moshe Y. Vardi, 'Automata-theoretic techniques for temporal reasoning', in *In Handbook of Modal Logic*, pp. 971–989, 2006.

[23] Moshe Y. Vardi and Larry J. Stockmeyer, 'Improved upper and lower bounds for modal logics of programs: Preliminary report', in *STOC*, pp. 240–251, 1985.

[24] Moshe Y. Vardi and Pierre Wolper, 'Automata-theoretic techniques for modal logics of programs', *J. Comput. Syst. Sci.*, **32**(2), 183–221, 1986.