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 $\mathrm{CTL}_{\mathcal{ALC}}$ and $\mathrm{CTL}^*_{\mathcal{ALC}}$ 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 $\mathrm{CTL}_{\mathcal{ALC}}$ and 2EXPTIME-completeness for satisfiability in $\mathrm{CTL}^*_{\mathcal{ALC}}$, thus the combined logics are computationally no more complex than their components.

As the next step, we stick with $\mathrm{CTL}_{\mathcal{ALC}}$ and $\mathrm{CTL}^*_{\mathcal{ALC}}$, 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 $\mathrm{CTL}_{\mathcal{ALC}}$ and a 3EXPTIME upper bound for $\mathrm{CTL}^*_{\mathcal{ALC}}$. For $\mathrm{CTL}_{\mathcal{ALC}}$, 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 $\mathrm{CTL}^*_{\mathcal{ALC}}$, 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 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 CTL$^*_{\mathcal{ALC}}$ and CTL$_{\mathcal{ALC}}$. Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be countably infinite sets of *concept names* and *role names*. CTL$^*_{\mathcal{ALC}}$*-state concepts* C and 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. CTL$_{\mathcal{ALC}}$ is the fragment of 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 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 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 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 $W \subseteq (\mathbb{N} \setminus \{0\})^*$ is a prefix-closed non-empty set of *nodes* and $E = \{(w, wc) \mid wc \in W, w \in \mathbb{N}^*, c \in \mathbb{N}\}$ 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 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 $c \in \mathbb{N}$ 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 $\pi = w_0w_1w_1 \cdots$ and $i \geq 0$, we use $\pi[i]$ to denote w_i and $\pi[i..]$ to denote the path $w_iw_{i+1} \cdots$. The mapping $\cdot^{\mathfrak{I},w}$ is extended from concept names to 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 CTL$^*_{\mathcal{ALC}}$-path concepts on a given path π, defined as:

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

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 CTL$_{\mathcal{ALC}}$ while the latter is CTL$^*_{\mathcal{ALC}}$ proper.

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

Our aim is to establish algorithms and tight complexity bounds for deciding satisfiability in CTL$_{\mathcal{ALC}}$- and 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 $\tau : T \rightarrow \Sigma$ 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 $\delta : Q \times \Sigma \rightarrow 2^{Q^k}$ 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 $\pi = w_0w_1 \cdots$ 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

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

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 $w \in (\mathbb{N} \setminus \{0\})^*$ 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 $c \in \mathbb{N}$. 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 satisfiability 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 $\mathcal{A}_{\overline{\varphi}} = (Q_1, \Sigma_1, \delta_1, Q^0_1, F_1)$ 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 $\mathcal{A}_\vartheta = (Q_2, \Sigma_2, \delta_2, Q^0_2, F_2)$ 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 $\mathbf{E}C$ 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 $\omega = (S_1, S_2)$:

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

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

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

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

where $\cdot$ in the third component means that the transition exists both when the component is t and f, and '$*$' behaves like a wildcard for all states of Q_2 with the test $* \in \{q_1, \dots, q_k\}$ 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$^{\mathbf{E}\Diamond,\mathbf{E}\Box}_{\mathcal{EL}}$ 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

$$X_1 \sqcap \dots \sqcap X_n \sqsubseteq X$$

with $X_1, \dots, X_n, X$ 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$^{\mathbf{E}\Diamond,\mathbf{E}\Box}_{\mathcal{EL}}$-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$^{\mathbf{E}\Diamond,\mathbf{E}\Box}_{\mathcal{EL}}$-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 $\text{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\Box}$, 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 $\text{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\Box,\mathbf{A}\Box}$ 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) $\text{CTL}_{\mathcal{EL}}^{\mathbf{A}\Diamond,\mathbf{E}\Diamond}$	*(b)* $\text{CTL}_{\mathcal{EL}}^{\mathbf{E}\Diamond,\mathbf{E}\bigcirc}$	*(c)* $\text{CTL}_{\mathcal{EL}}^{\mathbf{A}\Diamond,\mathbf{A}\bigcirc}$
(d) $\text{CTL}_{\mathcal{EL}}^{\mathbf{E}\mathcal{U}}$	*(e)* $\text{CTL}_{\mathcal{EL}}^{\mathbf{A}\mathcal{U}}$	*(f)* $\text{CTL}_{\mathcal{EL}}^{\mathbf{A}\bigcirc}$

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.

A Proofs for Section 3

By the following lemma, which is proved using standard automata-theoretic constructions, we can assume all states of $\mathcal{A}_\varphi$ (with φ the formula from Condition 2(c)) to be initial. We will from now on make this assumption throughout the appendix.

Lemma 5 *For every CTL*-formula $\mathbf{A}\Box\psi$ and the corresponding NBTA $\mathcal{A}_{\mathbf{A}\Box\psi} = (Q, \Sigma, \delta, Q_0, F)$, we can construct in time $O(\mathsf{poly}(|Q \times \Sigma|))$ an NBTA $\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi}$ such that $L(\mathcal{A}_{\mathbf{A}\Box\psi}) = L(\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi})$ and every state in $\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi}$ is an initial state.*

Proof. Let $\mathcal{A}' = (Q', \Sigma, \delta', Q'_0, F')$ be defined by setting $Q' = Q \times \Sigma$, $Q'_0 = Q_0 \times \Sigma$, and $F' = F \times \Sigma$. Intuitively, a state $(q, \sigma) \in Q'$ behaves just as q in $\mathcal{A}_{\mathbf{A}\Box\psi}$ when reading σ and rejects otherwise. More precisely, if $(q_1, \ldots, q_k) \in \delta(q, \sigma)$, then $((q_1, \sigma_1), \ldots, (q_k, \sigma_k)) \in \delta'((q, \sigma), \sigma)$ for all $\sigma_1, \ldots, \sigma_k \in \Sigma$. For all $\sigma' \neq \sigma$ we define $\delta'((q, \sigma), \sigma') = \emptyset$. It is easy to see that $\mathcal{A}_{\mathbf{A}\Box\psi}$ and $\mathcal{A}'$ accept the same language. Moreover, if r is an accepting run of $\mathcal{A}'$ on some structure (T, τ), then for all $w \in T$ we have

$$\tau(w) = \sigma \quad \Leftrightarrow \quad \exists q_w \in Q'.r(w) = (q_w, \sigma) \tag{1}$$

We call a state $q \in Q'$ *active* if there is some model (T, τ) of $\mathbf{A}\Box\psi$ that admits an accepting run r of $\mathcal{A}'$ such that $r(w) = q$ for some $w \in T$. Obviously, dropping inactive states does not change the language of $\mathcal{A}'$. So, let $\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi}$ be the variant of $\mathcal{A}'$ where $\widehat{Q}$ is the set of active states in Q' and $\widehat{Q}_0 = \widehat{Q}$, i.e., all states are initial states. We prove the following claim.

Claim. $L(\mathcal{A}') = L(\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi})$.
Proof of the claim. The direction "$\subseteq$" is immediate. For "$\supseteq$" assume that $\mathfrak{T} = (T, \tau) \in L(\widehat{\mathcal{A}}_\psi)$. Hence, there is an accepting run (T, r) of $\widehat{\mathcal{A}}_\psi$ on $\mathfrak{T}$. Let $r(\varepsilon) = (q, \sigma)$. Since, by assumption, (q, σ) is active, there is an accepting run r' of $\mathcal{A}'$ on some structure $\mathfrak{T}' = (T, \tau')$ with $r'(w) = (q, \sigma)$ for some $w \in T$. By Equation (1), $\tau'(w) = \sigma$. Let $\mathfrak{T}'' = (T, \tau'')$ the structure obtained from $\mathfrak{T}'$ be replacing the subtree rooted at w by $\mathfrak{T}$. It is straightforward to prove that $\mathfrak{T}''$ is accepted by $\mathcal{A}'$, hence $\mathfrak{T}''$ is a model of $\mathbf{A}\Box\psi$. Because $\mathbf{A}\Box\psi$ is a formula starting with $\mathbf{A}\Box$, the semantics implies that also the subtree of $\mathfrak{T}''$ rooted at w is a model of ψ. Hence, it is accepted by $\mathcal{A}'$. By construction, the subtree of $\mathfrak{T}''$ rooted at w is exactly $\mathfrak{T}$; thus, $\mathfrak{T}$ is accepted by $\mathcal{A}'$. This finishes the proof of the claim.

It remains to argue that we can check if a given state is active: Let $\mathcal{A}'' = (Q', Q' \times \Sigma, \delta'', Q'_0, F')$ be defined by

$$\delta''(q, (q', \sigma)) = \begin{cases} \delta'(q, \sigma) & \text{if } q = q' \\ \emptyset & \text{otherwise} \end{cases}$$

Intuitively, when reading a symbol (q, σ), a state q behaves in $\mathcal{A}''$ just as in $\mathcal{A}'$ when reading σ; it rejects when reading (q', σ) for $q' \neq q$. This implies that (T, r) is an accepting run of $\mathcal{A}'$ on some structure (T, τ) if and only if (T, r) is an accepting run of $\mathcal{A}''_\psi$ on (T, τ') where $\tau'(w) = (r(w), \tau(w))$ for all $w \in T$. We can easily devise an NBTA $\mathcal{B}_q$ that checks whether in $Q' \times \Sigma$-structure there is some world labeled with (q, σ) for some σ. Hence, a state q is active if and only if the language of $\mathcal{A}''_\psi \cap \mathcal{B}_q$ is not empty.

Since conjunction and emptiness check are polynomial time operations on Büchi tree-automata, this also yields that $\widehat{\mathcal{A}}_{\mathbf{A}\Box\psi}$ can be computed in polynomial time. ❑

In the appendix, we proceed in a slightly different order, i.e., we first give details on how to prove Lemma **??**, before we come to prove correctness of the algorithm (Lemma **??**).

Let S be the final result of type elimination. We extend S to a set $\widehat{S}$ that contains temporal types annotated with states of the automaton $\mathcal{A}_\varphi$. Specifically, an *extended type for $\mathcal{T}$* is a triple (t, q, i) with (t, i) a temporal type for $\mathcal{T}$ and $q \in Q$. Let $\widehat{S}$ be the set of all extended types (t, q, i) such that $(t, i) \in S$ and there is an 2^{cn}-labeled tree (T, τ) and an accepting run (T, r) of $\mathcal{A}_\varphi$ on (T, τ) such that

- for some $w \in T$ with $|w| = i$, we have $\tau(w) = \bar{t}$ and $r(w) = q$;
- for each $w \in T$, there is a $(t, \rho(i)) \in S$ with $\tau(w) = \bar{t}$.

Since S satisfies Condition 2, S is the projection of $\widehat{S}$ to the first and last component of triples. We observe the following.

Lemma 6 *For all $(t, q, i) \in \widehat{S}$, we have*

$\widehat{1}$. *if $\exists r.C \in t$, then there is a $(t', q', i) \in \widehat{S}$ such that $\{C\} \cup \{\neg D \mid \neg\exists r.D \in t\} \subseteq t'$;*
$\widehat{2}$. *there is a 2^{cn}-labeled tree (T, τ) and an accepting run (T, r) of $\mathcal{A}_\varphi$ on (T, τ) such that*
  (a) for some $w \in T$ with $|w| = i$, we have $\tau(w) = \bar{t}$ and $r(w) = q$;
  (b) for each $w \in T$ with $|w| = j$, there is a $(t', q', \rho(j)) \in \widehat{S}$ with $\tau(w) = \overline{t'}$ and $r(w) = q'$.

Proof. Condition $\widehat{1}$ is immediate. For condition $\widehat{2}$ observe that $(t, q, i) \in \widehat{S}$ because there is a 2^{cn}-labeled tree (T, τ) and an accepting run (T, r) of $\mathcal{A}_\varphi$ such that there is some $w^* \in T$ such with $|w^*| = i$, $\tau(w^*) = \bar{t}$ and $r(w^*) = q$ and for all $w \in T$ with $|w| = j$, there is $(t, \rho(j)) \in S$ with $\tau(w) = \bar{t}$. By definition of $\widehat{S}$, r and τ also witness that for all $w \in T$ we have $(t', r(w), \rho(|w|)) \in \widehat{S}$ where $\tau(w) = \bar{t}'$. Thus, τ and r together with w^* show that condition $\widehat{2}$ is satisfied. ❑

Let $\mathfrak{T}_i = \{(t, q) \mid (t, q, i) \in \widehat{S}\}$ for all $i \leq n_0$. We show the following monotonicity lemma:

Lemma 7 *For all $i \leq n_0$, we have*

1. $\mathfrak{T}_{i+1} \subseteq \mathfrak{T}_i$;
2. $\mathfrak{T}_i = \mathfrak{T}_{i+1}$ *implies* $\mathfrak{T}_i = \mathfrak{T}_{i+\ell}$ *for all* $i + \ell \leq n_0$

Proof. 1. Let $M = \widehat{S} \cup \{(t, q, j) \mid (t, q, i) \in \widehat{S}, \text{ for some } j \leq i\}$. We will show that M is consistent under the conditions $\widehat{1}$ and $\widehat{2}$. Let $(t, q, j) \in M$. By definition of M, either $(t, q, j) \in \widehat{S}$ or there is some $i \geq j$ such that $(t, q, i) \in \widehat{S}$. In the first case, it is immediate, so consider the second case.

- Lemma 6 implies that there is $(t', q', i) \in \widehat{S}$ witnessing condition $\widehat{1}$. Hence, $(t', q', j) \in M$ and (t, q, j) satisfies condition $\widehat{1}$.
- Lemma 6 implies that there is a 2^{cn}-labeled tree (T, τ) and an accepting run (T, r) of $\mathcal{A}_\varphi$ on (T, τ) such that (a) for some world $w \in T$ with $|w| = i$, we have $\tau(w) = \bar{t}$ and $r(w) = q$, and (b) for all $v \in T$ with $|v| = j$, there is a $(t', q', \rho(j)) \in \widehat{S}$ with $\tau(v) = \bar{t}'$ and $r(w) = q'$. Let $w = uv$ be such that $|v| = j$ and let (T, τ') be the subtree of (T, τ) rooted at w' and (T, r') be the subtree of r rooted at w'. Since all states of $\mathcal{A}_\varphi$ are initial states, (T, r') is an accepting run of $\mathcal{A}_\varphi$ on (T, τ'). By construction, $\tau'(v) = \bar{t}$ and $r'(v) = q$, hence condition $\widehat{2}(a)$ is satisfied. Also condition $\widehat{2}(b)$ is satisfied by definition of M.

2. Assume $\mathfrak{T}_i = \mathfrak{T}_{i+1}$ and let $M = \widehat{S} \cup \{(t,q,j) \mid (t,q,i) \in \widehat{S}$ and $i \leq j \leq n_0\}$. As in the previous case it is enough to check that all elements of M satisfy conditions $\widehat{1}$ and $\widehat{2}$. So let $(t,q,j) \in M$. By definition of M there is some $(t,q,i) \in \widehat{S}$ with $i \leq j$.

- By Lemma 6, there is some $(t',q',i) \in \widehat{S}$ witnessing condition $\widehat{1}$. Thus, $(t',q',j) \in M$ and (t,q,j) satisfies condition $\widehat{1}$.
- We show by induction on j that (t,q,j) is not eliminated. The cases $j = i$ and $j = i+1$ are trivial. For the induction step, assume $\mathfrak{T}_i = \mathfrak{T}_{i+\ell}$ for all $i+\ell < j$. By this assumption and the fact that $(t,q,i) \in M$, we know that $(t,q,j-1) \in M$. By condition $\widehat{2}$, there is a structure (T,τ) and an accepting run (T,r) of $\mathcal{A}_\varphi$ on (T,τ) such that (a) there is a world $w \in T$ with $|w| = j-1$ and $\tau(w) = \bar{t}$ and $r(w) = q$, and (b) for all $v \in T$ with $|v| = j$, there is a $(t',q',\rho(j)) \in \widehat{S}$ with $\tau(v) = \bar{t}'$ and $r(w) = q'$. Let $w = u \cdot c$ for some $c \in \mathbb{N}$ and $\tau u = t'$ and $r(u) = q'$. Since $|u| = j-2$, we have $(t',q',j-2) \in M$ and by hypothesis $(t',q',j-1) \in M$. By condition $\widehat{2}$, there is a structure (T,τ') and an accepting run $(T,r'$ of $\mathcal{A}_\varphi$ on (T,τ') such that for some $v \in T$ with $|v| = j-1$ such that $\tau'(v) = t'$ and $r'(v) = q'$. We define the tree (T,τ'') (the run (T,r''), respectively) to be the tree that is obtained from (T,τ') (from (T,r'), respectively) by replacing the subtree rooted at v by the subtree of (T,τ) (of (T,r), respectively) rooted at w'. Since $\tau'(v) = \tau(w')$ and $r'(v) = r(w')$, (T,r'') is an accepting run of $\mathcal{A}_\varphi$ on (T,τ''). By construction, $\tau''(v \cdot c) = t$ and $r''(v \cdot c) = q$. Thus, (T,τ'') and (T,r'') satisfy condition $\widehat{2}(a)$. By construction, they also satsify condition $\widehat{2}(b)$.

❑

Next, we define the infinite continuation of $\widehat{S}$ as $\widehat{S}_\omega = \{(t,q,i) \mid (t,q,\rho(i)) \in \widehat{S}\}$. Further, derive the conditions $\widehat{1}'$ and $\widehat{2}'$ from $\widehat{1}$ and $\widehat{2}$ by allowing all $i \in \mathbb{N}$ and replacing $\rho(i)$ by i in $\widehat{2}(b)$.

Lemma 8 *Every $(t,q,i) \in \widehat{S}_\omega$ satisfies conditions $\widehat{1}'$ and $\widehat{2}'$.*

Proof. Since by the choice of n_0, $|\mathfrak{T}_0| \leq n_0$, Lemma 7 implies that either $\mathfrak{T}_{n_0} = \emptyset$ or $\mathfrak{T}_{n_0} = \mathfrak{T}_{n_0-1}$. In the first case $\widehat{S}_\omega = \emptyset$ and we are done. In the second case, we show by induction that over $i \geq 0$ that all triples (t,q,n_0+i-1) satisfiy conditions $\widehat{1}'$ and $\widehat{2}'$. The induction base for $i = 0$ and $i = 1$ is trivial. For the induction step assume $(t,q,i) \in \widehat{S}_\omega$ and that all (t',q',ℓ) satisfy $\widehat{1}'$ and $\widehat{2}'$ for $\ell < i$. Now, we can proceed as in part 2 of the proof of Lemma 7 to show that (t,q,i) satisfies conditions $\widehat{1}'$ and $\widehat{2}'$. ❑

So Lemma 8 justifies the choice of n_0. Finally, we come back to the Lemma **??**.

Lemma ?? *If S is the result of type elimination and $(t,n_0) \in S$ is S-realizable, then $(t,n_0+\ell)$ is S-realizable, for any $\ell \geq 0$.*

Proof. We define conditions 1' and 2' as variants of condition 1 and 2 by admitting every $i \in \mathbb{N}$ and replacing $\rho(i)$ with i in condition 2. Observe that $S_\omega = S \cup \{(t,m) \mid (t,n_0) \in S, m > n_0\}$ as defined in the paper is precisely the projection of $\widehat{S}_\omega$ to the first and third component of the triples. So, let $(t,n_0) \in S$ be S-realizable. By definition of $\widehat{S}$, there is some q such that $(t,q,n_0) \in \widehat{S}$. By Lemma 8, $(t,q,n_0+\ell) \in \widehat{S}_\omega$ for every $\ell \geq 0$, i.e., $(t,q,n_0+\ell)$ satisfies conditions $\widehat{1}'$ and $\widehat{2}'$. Thus, there is some $(t',q',n_0+\ell) \in \widehat{S}_\omega$ witnessing condition $\widehat{1}'$. By definition of $\widehat{S}_\omega$, $(t',q',n_0) \in S_\omega$, thus $(t,n_0) \in S$ and $(t,n_0+\ell) \in S_\omega$. Hence, condition 1' is satisfied for $(t,n_0+\ell)$. Analogously, it can be shown that $(t,n_0+\ell)$ satisfies condition 2'. Hence, it is S-realizable. ❑

Lemma ?? *The algorithm returns "satisfiable" iff $\mathcal{T}$ is satisfiable.*

Proof.
"⇒": Let S_ω be the result of the type elimination procedure. In the following fix $k = \sharp_{\mathbf{E}}(\mathcal{T})$ and let T the complete k-ary tree. Due to condition 2', for every $(t,i) \in S_\omega$ there is a 2^{cn}-labeled k-ary tree $(T,\tau_{t,i})$ that is a model for φ. Define the temporal interpretation $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w \in T})$ by taking $\Delta = S_\omega$ and

$$\begin{array}{rcl} A^{\mathfrak{I},w} & = & \{(t,i) \mid A \in \tau_{t,i}(w)\} \\ r^{\mathfrak{I},w} & = & \{(t,i),(t',i) \mid \exists r.C \in \tau_{t,i}(w) \text{ implies} \\ & & \quad \{A\} \cup \{\neg E \mid \neg\exists r.E \in \tau_{t,i}(w)\} \subseteq \tau_{t',i}(w)\} \end{array}$$

Since condition 1' is satisfied for every $(t,i) \in S_\omega$, $\mathfrak{I}$ is a valid temporal model. Now, by definition, $C_\mathcal{T} \in t$ for all $(t,i) \in S_\omega$. Thus $X_{C_\mathcal{T}} \in \bar{t}$ for all $(t,i) \in S_\Omega$ and φ ensures that $\overline{C}_\mathcal{T}$ is satisfied in every point. Hence $\mathfrak{I}$ is a model of $\mathcal{T}$.

"⇐": Let $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w \in W})$ a model of $\mathcal{T}$. Define for every $d \in \Delta$ the $2^{\mathsf{cl}(\mathcal{T})}$-labeled tree (T,τ_d) by

$$\tau_d(w) = \{C \in \mathsf{cl}(\mathcal{T}) \mid d \in C^{\mathfrak{I},w}\}$$

and the 2^{cn}-labeled tree $\overline{\tau}_d$ by $\overline{\tau}_d(w) = \overline{\tau_d(w)}$ for all $w \in T$. Now define

$$S = \{(\tau_d(w), i) \mid w \in T, d \in \Delta, i \leq \rho(|w|)\}$$

It remains to verify that every $(t,i) \in S$ is S-realizable. Condition 1 is immediately satisfied by definition of S. For condition 2 let $(t,i) \in S$, i.e., there is some $w \in T$, $d \in \Delta$ such that $t = \tau_d(w)$ and $i \leq \rho(|w|)$. Now, let $w = uv$ with $|v| = i$ (possible, since $i \leq \rho(|w|)$ and thus $i \leq |w|$). It can be shown that the subtree of $(T,\overline{\tau}_d)$ rooted at u satifies precisely the requirements of condition 2. ❑

B Proofs for Section 4

Proposition 2 *Let φ be a temporal CTL$^*_{\mathcal{ALC}}$-TBox. φ is satisfiable if and only if there exists a proper quasi-model of φ.*

Proof. ⇒: Let $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w \in W})$ be a temporal model of φ. We define a $\mathsf{q(w)}$-labeled tree structure $\mathfrak{M} = (T,\tau)$ such that for all $w \in T$, $\tau(w)$ is defined as follows:

$$\begin{array}{rcl} S_2(w) & = & \{\Psi \in \mathsf{sub}(\varphi) \mid \mathfrak{I}, w \models \Psi\} \\ \pi(d,w) & = & \{C \in \mathsf{cl}(\varphi) \mid d \in C^{\mathfrak{I},w}\} \\ S_1(w) & = & \{\pi(d,w) \mid d \in \Delta\}; \end{array}$$

We obtain the $\underline{2^{\mathsf{cn}_Y}}$- labeled tree $\mathfrak{M}_2$ by associating each $w \in \mathfrak{M}$ with the label $S_2(w)$. Moreover, it is clear that for all $w \in T$ with $\tau(w) = (S_1, S_2)$ and all $\pi(d,w) \in S_1$ there is a 2^{cn_X}-labeled tree (T,τ') satisfying 2(a)-(c). Then, $\mathfrak{M}$ is indeed a proper-quasimodel of φ.

⇐: Let $\mathfrak{M} = (T,\tau)$ be a proper-quasimodel of φ. According to Condition 2 , for all $w \in T$ with $\tau(w) = (S_1,S_2)$ and all $s \in S_1$ there is a there is a 2^{cn_X}-labeled tree $(T,\tau_{w,s})$ satisfying 2(a)-(c). We define the temporal interpretation $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w \in W})$ with $\Delta = \{(w,s) \mid s \in S_1(w)\}$ given by:

$$\begin{array}{rcl} A^{\mathfrak{I},w} & = & \{(v,s) \in \Delta \mid A \in \tau_{v,s}(w)\} \text{ for all } A \in \mathsf{N_C} \\ r^{\mathfrak{I},w} & = & \{((v,s),(v',s')) \mid \exists r.A \in \mathcal{L}_{v,s}(w) \text{ implies} \\ & & \{A\} \cup \{\neg E \mid \neg\exists r.E \in \tau_{v,s}(w)\} \subseteq \tau_{v',s'}(w)\} \\ & & \text{for all } r \in \mathsf{N_R}. \end{array}$$

By using the properties of a proper-quasimodel we can prove that $\mathfrak{I}$ is a temporal model of φ. ❑

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

Proof.

The "if"-direction is trivial. For the other direction let $\mathfrak{M} = (T, \tau)$ be an arbitrary proper quasi-model. We can assume that every $w \in T$ has outdegree at least k, since otherwise we can just duplicate some successors of w.

Now, we modify Condition 2 by restating 2(c) in terms of automata:

2.' for all $w \in T$ with $\tau(w) = (S_1, S_2)$ and all $s \in S_1$, there is a $2^{\mathsf{cn}X}$-labeled $\sharp^{\mathsf{c}}_E$-ary tree (T', τ') with T constructed from T 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) There is an accepting run (T', r) of $\mathcal{A}_\vartheta$ on (T', τ').

This is sufficient, because if there is a tree-shaped model of ϑ, then there is one with branching degree $\sharp^{\mathsf{c}}_E$; and $L(\mathcal{A}_\vartheta)$ is precisely $\mathsf{Mod}_{\sharp^{\mathsf{c}}_E}(\vartheta)$. In the following we denote with "$s \in \tau(w)$" the fact that $s \in S_1$ when $\tau(w) = (S_1, S_2)$. Fix for every $w \in T$, $s \in \tau(w)$:

- the tree $(T_{w,s}, \tau_{w,s})$ witnessing condition 2',
- the corresponding run $r_{w,s}$ from 2'(c), and
- the set of states $Q_{w,s}$ that $\mathcal{A}_\vartheta$ assigns to s in *all* accepting runs, i.e.,

$$Q_{w,s} = \{r_{w',s'}(w) \mid w \in T_{w',s'}, \tau_{w',s'}(w) = \overline{s}\}$$

Now, we duplicate quasi-worlds of $\mathfrak{M}$ and modify the trees $(T_{w,s}, \tau_{w,s})$ and the corresponding runs $r_{w,s}$ such that at most precisely one type in every quasi-world is assigned a state from some run, i.e., after the modification it is

$$\sum_{s \in \tau(w)} |Q_{w,s}| = 1$$

for every $w \in T$. More formally, we define the quasi-model $\widehat{\mathfrak{M}} = (\widehat{T}, \widehat{\tau})$ with $\widehat{T} = (\widehat{W}, \widehat{E})$ by taking

$$\begin{aligned}
\widehat{W} &= \{(w, s, q) \mid w \neq \varepsilon \in T, s \in \tau(w), q \in Q_{w,s}\} \\
&\quad \cup \{\varepsilon\} \\
\widehat{E} &= \{((w, s, q), (w', s', q')) \mid (w, w') \in E\} \cup \\
&\quad \{(\varepsilon, (w, s, q)) \mid (\varepsilon, w) \in E\} \\
\widehat{\tau}(w, s, q) &= \tau(w) \\
\widehat{\tau}(\varepsilon) &= \tau(\varepsilon)
\end{aligned}$$

Note that, by definition of $\widehat{\tau}$, Condition 1 is still satisfied for $\widehat{\mathfrak{M}}$. We make use of $T_{w,s}$, $\tau_{w,s}$, and $r_{w,s}$ to show that $\widehat{\mathfrak{M}}$ satisfies also Condition 2'. For all $(w, s, q) \in \widehat{T}$ and $t \in \widehat{\tau}(w, s, q)$ define

$$\begin{aligned}
T_{(w,s,q),t} &= \{(w', \tau_{w,t}(w'), r_{w,t}(w')) \mid w' \in T_{w,t} \setminus \{w\}\} \\
&\quad \cup \{(w, s, q)\} \\
\tau_{(w,s,q),t}(w') &= \tau_{w,t}(w') \\
r_{(w,s,q),t}(w') &= r_{w,t}(w')
\end{aligned}$$

We argue now that for each $w, w_1, w_2 \in T$ and $(w, w_1) \in E$, $(w, w_2) \in E$ and $\tau(w_1) = \tau(w_2)$, we can remove either the subtree $(\widehat{T}, \widehat{\tau}')$ of $\widehat{M}$ rooted at (w_1, s, q) or the subtree $(\widehat{T}, \widehat{\tau}'')$ of $\widehat{M}$ rooted at (w_2, s, q) from $\widehat{\mathfrak{M}}$. Without loss of generality, let us remove $(\widehat{T}, \widehat{\tau}'')$. We show that the resulting $\widehat{\mathfrak{M}}' = (\widehat{T}', \widehat{\tau}')$ continues satisfying condition $2'$.

- We use w' to denote a node of the form (w', s', q'). For every $(T_{w',t}, \tau_{w,t})$ that contains (w_2, s, q) we can construct a $(T_{w',t}, \tau'_{w',t})$ by replacing the subtree of $(T_{w',t}, \tau_{w,t})$ rooted in (w_2, s, q) by an appropriate $(T_{w',t}, \tau''_{w',t})$ with root (w_1, s, q). More formally,

$$\tau'_{w',t}(w) = \begin{cases} \tau''_{w',t}(w) & \text{if } w \in (T_{w',t}, \tau''_{w',t}) \\ \tau_{w',t}(w) & \text{otherwise} \end{cases}$$

$$r'_{w',t}(w) = \begin{cases} r''_{w',t}(w) & \text{if } w \in (T_{w',t}, \tau''_{w',t}) \\ r_{w',t}(w) & \text{otherwise} \end{cases}$$

Now, it is not hard to see that $\widehat{M}'$ satisfies the condition 2', and that it it is a k-ary tree.

❑

Lemma 9 *For a temporal CTL$^*_{\mathcal{ALC}}$ TBox φ, one can construct a 2ABTA $\mathcal{A} = (Q, \Sigma, \delta, \{q_0\}, F)$ where $\Sigma = 2^{\mathsf{qw}(\varphi)}$, $|Q| \in 2^{2^{\mathsf{poly}(|\varphi|)}}$ and $|Q| \in 2^{\mathsf{poly}(|\varphi|)}$ when φ is a temporal CTL$_{\mathcal{ALC}}$ temporal TBox such that $L(\mathcal{A})$ is the set of proper-quasimodels satisfying φ.*

Proof. Using k from Lemma 3, we construct $\mathcal{A}$ on k-ary $\mathsf{qw}(\varphi)$-labeled trees. $\mathcal{A}$ is constructed as presented in Section 4, i.e., we construct separately 2ABTAs $\mathcal{A}_1$ and $\mathcal{A}_2$ to deal with conditions 1 and 2 of proper quasimodels. To construct $\mathcal{A}_1$ and verify its correctness is straightforward (cf. Section 4). We proceed to check the correctness of $\mathcal{A}_2 = (Q, \Sigma, \delta, \{q_0\}, F)$ checking condition 2:

1. for all $w \in T$ with $\tau(w) = (S_1, S_2)$ and all $s \in S_1$, there is a $2^{\mathsf{cn}X}$-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 $\vartheta = \mathbf{A}\Box \bigwedge_{X_C \in \mathsf{cn}X} (X_C \leftrightarrow \overline{C})$.

$\mathcal{A}_2$ verifies Condition 2 by simulating a run of $\mathcal{A}_\vartheta = (Q_2, \Sigma_2, \delta_2, Q^0_2, F_2)$ for every $w \in T$ with $\tau(w) = (S_1, S_2)$ and every $s \in S_1$ where $\mathcal{A}_\vartheta$ is the NBTA on $2^{\mathsf{cn}X}$-trees accepting the models of ϑ.

$\mathcal{A}_2$ is sound. To check that given an accepting run (T_r, r) on a k-ary $\mathsf{qw}(\varphi)$-labeled tree $\mathfrak{M} = (T, \tau)$, then $\mathfrak{M}$ fulfils condition 2. Recall that a run on $\mathfrak{M}$ is a $T \times Q$-labeled tree. Then, it is not hard to see that with the labellings $\tau_r(x) = (w, q)$ for q of the form (q, s) with $q \in Q_2, s \in 2^{\mathsf{cn}X}$, using the second component, we can construct for all $w \in T$ with $\tau(w) = (S_1, S_2)$ and all $s \in S_1$ a $2^{\mathsf{cn}X}$-labeled tree (T, τ') satisfying conditions 2(a)-(b). Moreover, the first component provide us with an accepting run of $\mathcal{A}_\vartheta$ on (T, τ'). In particular, note that $F = F_2$, and as discussed in Section 4 we define adequately predecessor states to ensure that the accepting run of $\mathcal{A}_\vartheta$ starts at ε. Hence, 2(c) is satisfied.

$\mathcal{A}_2$ is complete. To check that a given a proper-quasimodel $\mathfrak{M} = (T, \tau)$ is accepted by $\mathcal{A}_2$. This is, there is an accepting run (T_r, τ_r) of $\mathcal{A}_2$ on $\mathfrak{M}$. Let $\mathfrak{M}_{w,s} = (T, \tau)$ be a $2^{\mathsf{cn}X}$-labeled tree for some $w \in T, s \in S_1$ satisfying 2. Due to 2(c) there is an accepting run (T, r) of $\mathcal{A}_\vartheta$ on $\mathfrak{M}_{w,s}$. Since $\mathcal{A}$ simulates the runs of $\mathcal{A}_\vartheta$, its clear that we can use r to define a partial run for $\mathcal{A}$. By using the accepting runs of $\mathcal{A}_\vartheta$ on all $\mathfrak{M}_{w',s'}$ we can define a run for $\mathcal{A}_2$. Moreover $F = F_2$, and thus (T_r, τ_r) is an accepting run on $\mathfrak{M}$.

The number of states $\mathcal{A}_\vartheta$ is in $O(2^{2^{\mathsf{poly}(\vartheta)}})$, and $O(2^{\mathsf{poly}(\vartheta)})$ if we consider CTL$_{\mathcal{ALC}}$-TBoxes. Then,

$$Q = \{q_0\} \cup Q_1 \cup (Q_1 \times 2^{\mathsf{cn}\,X}) \cup (Q_1 \times Q_1) \cup (Q_1 \times 2^{\mathsf{cn}\,X} \times Q_1)$$

is in $O(2^{2^{\mathsf{poly}(\varphi)}})$ ($O(2^{\mathsf{poly}(\varphi)})$ if we consider CTL$_{\mathcal{ALC}}$-TBoxes). ❑

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

Proof. By Lemma 9 $\mathcal{A}_\varphi$ has $O(2^{2^{\mathsf{poly}(\varphi)}})$ states if φ is a temporal CTL$^*_{\mathcal{ALC}}$ TBox, and $(O(2^{\mathsf{poly}(\varphi)})$ if φ is a temporal CTL$_{\mathcal{ALC}}$ TBox, and the emptiness for 2ABTA can be decided in exponential time in the number of states. Therefore, satisfiability of temporal TBoxes is in 2EXPTIME for CTL$_{\mathcal{ALC}}$ and in 3EXPTIME for CTL$^*_{\mathcal{ALC}}$ ❑

Lemma 10 *Satisfiabilty of temporal* CTL$_{\mathcal{ALC}}$ *TBoxes is* 2-EXPTIME-*hard*

Proof. The proof is by reduction of the word problem for exponentially space alternating Turing machines. An ATM is a tuple $\mathcal{M} = (Q, \Sigma, \Gamma, q_0, \delta)$, where:

- Q is a set of states containing pairwise disjoint sets of *existential states* $Q_\exists$, *universal states* $Q_\forall$, and *halting states* $\{q_a, q_r\}$, where q_a is an *accepting* and q_r a *rejecting* state;
- Σ is an *input alphabet* and Γ a *working alphabet*, containing the *blank symbol* $\llcorner\!\lrcorner$ such that $\Sigma \subseteq \Gamma$ and $\llcorner\!\lrcorner \notin \Sigma$;
- $q_0 \in Q_\exists \cup Q_\forall$ is the *initial state*;
- δ is a *transition relation* is of the form $\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{\ell, r, n\}$. We write $(q', b, m) \in \delta(q, a)$ for $(q, a, q', b, m) \in \delta$. We assume that $q \in Q_\exists \cup Q_\forall$ implies $\delta(q, b) \neq \emptyset$ for all $b \in \Gamma$ and $q \in \{q_a, q_r\}$ implies $\delta(q, b) = \emptyset$ for all $b \in \Gamma$. Intuitively, the triple (q', b, m) describes the transition to state q', involving overwriting of symbol a with b and a shift of the head to the left ($m = l$), to the right ($m = r$) or no shift ($m = n$).

A *configuration* of an ATM is a word wqw' with $w, w' \in \Gamma^*$ and $q \in Q$ stating that the tape contains the word ww' (with only blanks before and behind it), the machine is in state q, and the head is on the leftmost symbol of w'. The *successor configurations* of a configuration wqw' are defined in terms of the transition relation δ. A *halting configuration* is of the form wqw' with $q \in \{q_a, q_r\}$.

A *computation path* of an ATM $\mathcal{M}$ on a word w is a (finite or infinite) sequence of configurations $c_1, c_2, \ldots$ such that $c_1 = q_0 w$ and c_{i+1} is a successor configuration of c_i for $i \geq 0$. All ATMs considered in this paper have only *finite* computation paths on any input[2]. A halting configuration is *accepting* iff it is of the form $wq_a w'$. A non-halting configurations $c = wqw'$ is accepting if at least one (all) successor configurations is accepting for $q \in Q_\exists$ ($q \in Q_\forall$, respectively). An ATM *accepts* an input w if the *initial configuration* $q_0 w$ is accepting. We denote $L(\mathcal{M})$ the language $\{w \in \Sigma^* \mid \mathcal{M} \text{ accepts } w\}$.

We set the configurations of an accepting computation of an ATM $\mathcal{M}$ on a word w in an *acceptance tree* which is a finite tree whose nodes are labelled with configurations such that

- the root node is labelled with the initial configuration $q_0 w$;
- if a node s in the tree is labelled with wqw', $q \in Q_\exists$, then s has exactly one successor, and this successor is labelled with a successor configuration of wqw';
- if a node s in the tree is labelled with wqw', $q \in Q_\forall$, then there is exactly one successor of s for each successor configuration of wqw';
- leaves are labelled with accepting halting configurations.

According to [10], the problem of deciding whether $w \in L(\mathcal{M})$ is 2-EXPTIME-hard. We assume that the length of every computation of $\mathcal{M}$ on $w \in \Sigma^k$ is bounded by 2^{2^k}, and for all configurations uqu' in this computation $\mid uu' \mid \leq 2^k$.

Let $\mathcal{M} = (Q, \Sigma, \Gamma, q_0, \delta)$ be an ATM and $w = \sigma_0, \ldots, \sigma_{k-1}$ the word for which we want to decide whether $w \in L(\mathcal{M})$, then we construct a temporal CTL$_{\mathcal{ALC}}$ TBox $\mathcal{T}_{\mathcal{M},w}$ such that $w \in L(\mathcal{M})$ *iff* $\mathcal{T}_{\mathcal{M},w}$ is satisfiable. In a model of $\mathcal{T}_{\mathcal{M},w}$ an accepting computation of $\mathcal{M}$ is identified with a (temporal) tree structure of the model. In this way each time point of the tree is associated with a single tape cell of a configuration of $\mathcal{M}$, and clearly going to a successor leads to next tape cell in the current configuration or to the first cell of the successor configuration. The temporal TBox $\mathcal{T}_{\mathcal{M},w}$ is formed by the conjunction of several formulas. First, we introduce formulas that allow us to establish the basic features of the tree structure.

- There exists always a time successor until we reach the head in a halting configuration.

$$\mathbf{A}\Box(\neg(Q_a \sqcup Q_r) \sqsubseteq \mathbf{E}\bigcirc \top) \tag{2}$$

- Every time point has globally associated the alphabet letter of a tape cell.

$$\mathbf{A}\Box(\bigvee_{a \in \Gamma} \top \sqsubseteq A_a) \tag{3}$$

- Each tape cell is labelled with exactly one alphabet letter.

$$\mathbf{A}\Box(\top \sqsubseteq \bigsqcap_{a,a' \in \Gamma, a \neq a'} \neg(A_a \sqcap A_{a'})) \tag{4}$$

- We use several counters over the temporal tree structure for transversing it. Each counter consists of a number of inclusions of polynomial size. In particular, a counter permits us to identify time points on the branches at a fixed distance 2^k. Constraints (5)-(9) implement an exemplary *counter*, based on atomic concepts X_i for $1 \leq i \leq k$, which simulate the bits of a number in binary .

$$(\mathit{Count}^X = 0) \equiv \bigsqcap_{j=1}^{n} \neg X_j, \tag{5}$$

$$\neg X_i \sqcap \neg X_j \sqsubseteq \mathbf{A}\bigcirc \neg X_i, \text{ for every } 1 \leq j < i \leq k, \tag{6}$$

$$X_i \sqcap \neg X_j \sqsubseteq \mathbf{A}\bigcirc X_i, \text{ for every } 1 \leq j < i \leq k, \tag{7}$$

$$\neg X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 \sqsubseteq \mathbf{A}\bigcirc X_j, \text{ for every } 1 \leq j \leq k, \tag{8}$$

$$X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 \sqsubseteq \mathbf{A}\bigcirc \neg X_j, \text{ for every } 1 \leq j \leq k. \tag{9}$$

We instantiate the above pattern to encode different counters X, e.g., Count^{tape} denotes a tape counter. We use $\mathit{Count}^X = N$ as the obvious abbreviation for setting the counter X to the value N. Further, we introduce the abbreviations End^X and Start^X to denote $\mathit{Count}^X = 2^k - 1$ and $\mathit{Count}^X = 0$, respectively. Now, we enforce the conditions that ensure that the model actually represents an accepting computation of $\mathcal{M}$ on w. First, we introduce the following auxiliary concept names

– H to mark the tape cells right of the head;

[2] As this case is simpler than the general one, we define acceptance for ATMs with finite computation paths only, and refer to [10] for the full definition.

- Q_q for every $q \in Q$;
- $M_{q,a,m}$ for every $(q,a,m) \in \Theta = \{(q,a,m) \mid (q',b,q,a,m) \in \delta$ for any $b \in \Gamma$ and $q' \in Q\}$;
- $Count^{tape}$.

- In each configuration every tape cell is labelled with at most one state variable Q_q:

$$\mathbf{A}\Box(\top \sqsubseteq \bigsqcap_{q,q' \in Q} \neg(Q_q \sqcap Q_{q'})) \quad (10)$$

- H marks the tapes cells that are to the right of the head in the current configuration.

$$\mathbf{A}\Box(\bigsqcup_{q \in Q} Q_q \sqsubseteq \mathbf{A}\bigcirc H) \quad (11)$$

$$\mathbf{A}\Box(H \sqcap \neg End^{tape} \sqsubseteq \mathbf{A}\bigcirc H) \quad (12)$$

- In every configuration at most one cell is labelled with a state:

$$\mathbf{A}\Box(H \sqsubseteq \neg \bigsqcup_{q \in Q} Q_q) \quad (13)$$

- $M_{q,a,m}$ concepts serve for carrying the information generated by the transition function, and use to determine the successor configuration. Information about the transitions is generated depending on whether the state is universal (14) or existential (15) and then carried to the end of the tape (16). Moreover, we initialize a counter (implemented with concept names R_i) marking the position of the head (14) - (15).

$$\mathbf{A}\Box(A_a \sqcap Q_q \sqsubseteq \bigsqcap_{(q'b'm) \in \delta(q,a)} M_{q',b,m} \sqcap Start^{head}) \quad (14)$$
$$\text{for every } a \in \Gamma, q \in Q_\forall$$

$$\mathbf{A}\Box(A_a \sqcap Q_q \sqsubseteq \bigsqcup_{(q'b'm) \in \delta(q,a)} M_{q',b,m} \sqcap Start^{head}) \quad (15)$$
$$\text{for every } a \in \Gamma, q \in Q_\exists$$

$$\mathbf{A}\Box(M_{q,a,m} \sqcap \neg End^{tape} \sqsubseteq \mathbf{A}\bigcirc M_{q,a,m}) \quad (16)$$

- When moving to a successor configuration in order to avoid clashes in the information we create copies $N_{q,a,m}$ for concepts $M_{q,a,m}$ (17)-(18).

$$\mathbf{A}\Box(M_{q,a,m} \sqcap End^{tape} \sqsubseteq \mathbf{E}\bigcirc N_{q,a,m}) \quad (17)$$
$$\text{for every } (q,a,m) \in \Theta$$

$$\mathbf{A}\Box(N_{q,a,m} \sqcap \neg End^{tape} \sqsubseteq \mathbf{A}\bigcirc N_{q,a,m}) \quad (18)$$

- At most one concept $N_{q,a,m}$ is true in a tape cell

$$\mathbf{A}\Box(\top \sqsubseteq \bigsqcap_{(q,a,m),(q',a',m') \in \Theta} \neg(N_{q,a,m} \sqcap N_{q',a',m'})) \quad (19)$$

- To avoid further clashes while synchronizing adjacent configurations, we create a copy (20)-(21) of the head counter $Count^{head}$, $Count^{head'}$ implemented with concepts R'_i.

$$\mathbf{A}\Box(End^{tape} \sqcap R_i \sqsubseteq R'_i) \quad (20)$$

$$\mathbf{A}\Box(End^{tape} \sqcap \neg R_i \sqsubseteq \neg R'_i) \quad (21)$$

- Changes imposed by the transition relation are implemented: write the new tape symbol (22), and place the state variable in the correct position (23) - (25). The transition (26) - (27) does not push the head beyond the tape.

$$\mathbf{A}\Box(N_{q,a,m} \sqcap Start^{head'} \sqsubseteq A_a), \text{ for every } (q,a,m) \in \Theta \quad (22)$$

$$\mathbf{A}\Box(N_{q,a,m} \sqcap Start^{head'} \sqsubseteq Q_q), \text{ for every } (q,a,n) \in \Theta \quad (23)$$

$$\mathbf{A}\Box(N_{q,a,m} \sqcap Start^{head'} \sqsubseteq \mathbf{A}\bigcirc Q_q), \text{ for every } (q,a,r) \in \Theta \quad (24)$$

$$\mathbf{A}\Box(N_{q,a,m} \sqcap End^{head'} \sqsubseteq Q_q), \text{ for every } (q,a,l) \in \Theta \quad (25)$$

$$\mathbf{A}\Box(Start^{head'} \sqcap Start^{tape} \sqsubseteq \neg N_{q,a,m}), \text{ for every } (q,a,l) \in \Theta \quad (26)$$

$$\mathbf{A}\Box(End^{head'} \sqcap End^{tape} \sqsubseteq \neg N_{q,a,m}), \text{ for every } (q,a,r) \in \Theta \quad (27)$$

- We propagate the information of each i-th tape cell that do not change during the transition. This information is store in fresh elements, i.e., new r-successors, and synchronized via the counter $Counter^{cell}$ with the content of the i-th cell in the previous configuration. To store the labelled of a tape cell we introduce concept names W_a, S_a for every $a \in \Gamma$.
  - The information is spread through the tape to the previous configuration.

$$\mathbf{A}\Box(\neg End^{tape} \sqcap \mathbf{E}\bigcirc W_a \sqsubseteq W_a) \text{ for every } a \in \Gamma \quad (28)$$

$$\mathbf{A}\Box(\neg End^{tape} \sqcap \mathbf{E}\bigcirc S_a \sqsubseteq S_a) \text{ for every } a \in \Gamma \quad (29)$$

  - An element stores exactly on alphabet letter.

$$\mathbf{A}\Box(\top \sqsubseteq \bigsqcap_{a,a' \in \Gamma, a \neq a'} \neg(W_a \sqcap W_{a'})) \quad (30)$$

$$\mathbf{A}\Box(\top \sqsubseteq \bigsqcap_{a,a' \in \Gamma, a \neq a'} \neg(S_a \sqcap S_{a'})) \quad (31)$$

  - S_a is used as a copy of the concept W_a in the previous configuration.

$$\mathbf{A}\Box(End^{tape} \sqcap \mathbf{E}\bigcirc W_a \sqsubseteq S_a) \text{ for every } a \in \Gamma \quad (32)$$

  - A representative of each cell (not meant to change) is generated and its label A_a is store W_a

$$\mathbf{A}\Box((\neg Start^{head'} A_a \sqsubseteq \exists r.W_a \sqcap Start^{cell}) \sqcap \quad (33)$$

  - We synchronize the content of i-th cell with that of the i-th cell in the previous configuration.

$$\mathbf{A}\Box(A_a \sqcap S_b \sqcap Start^{cell} \sqsubseteq \bot), \text{ for every } b \neq a \in \Gamma. \quad (34)$$

- The input $w = \sigma_0, \dots \sigma_{k-1}$ is accepted. Recall that in our setting any computation is terminating, moreover halting configurations are the only configurations without successor configurations. Then, the input is accepted if the rejecting state is not reached.

$$\mathbf{A}\Box(\top \sqsubseteq \neg Q_{q_r}) \quad (35)$$

- The initial configuration $q_0 w$starting at A_0 is encoded as follows

$$\begin{array}{rcl} A_0 & = & A_{a_1} \sqcap Q_{q_0} \sqcap Start^{tape} \sqcap \mathbf{E}\bigcirc A_1 \\ A_i & = & A_{a_{i+1}} \sqcap \mathbf{E}\bigcirc A_{i+1} \\ A_n & = & \mathbf{A}((A_\llcorner \sqcap \neg End^{tape})\mathcal{U}(A_\llcorner \sqcap End^{tape})) \end{array}$$

$$\neg(\top \sqsubseteq \neg A_0) \qquad (36)$$

The definition of $\mathcal{T}_{\mathcal{M},\omega}$ is the conjunction of the TBoxes introduced above. Now, it is not hard to see that the size of $\mathcal{T}_{\mathcal{M},\omega}$ is polynomial in k. Finally, following the intuitive meaning of each conjunct given above, it is also not hard to check that $\mathcal{T}_{\mathcal{M},\omega}$ is satisfiable iff $w \in L(\mathcal{M})$. ❑

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

Proof. The upper bound follows from Lemma 9, and the lower bound from Lemma 10. ❑

C Proofs for Section 5

Lemma 4 *Let A_0, B_0 two concepts ocurring in $\mathcal{T}$. Then, $A_0 \sqsubseteq_{\mathcal{T}} B_0$ iff $A_0 \sqsubseteq_{\mathcal{T}'} B_0$.*

Proof. We use $\hat{C}$ to denote the translation introduced in Section 5, i.e., $\mathbf{E}\hat{\Diamond}A = \exists \mathsf{succ}_\Diamond.A$ and $\mathbf{E}\hat{\Box}A = M_A$ for some fresh concept name M_A.

$\Rightarrow$) We show the contrapositive, i.e., $\mathcal{T}' \models A_0 \not\sqsubseteq B_0$, then $\mathcal{T} \models A_0 \not\sqsubseteq B_0$. $\mathcal{T}' \models A_0 \not\sqsubseteq B_0$ if and only if there is a model $\mathcal{I}$ of $\mathcal{T}'$ such that there is a $x \in A_0^{\mathcal{I}}$, but $x \notin B_0^{\mathcal{I}}$. Then, we construct a temporal model $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w\in W})$ of $\mathcal{T}$ based on $\mathcal{I}$ such that $x \in A_0^{\mathfrak{I},\varepsilon}$, but $x \notin B_0^{\mathfrak{I},\varepsilon}$. From now on, w.l.o.g. we assume that $\mathcal{I}$ is tree shaped.

We define sequences $\Delta_0, \Delta_1, \ldots, W_0, W_1, \ldots, E_0, E_1, \ldots$, and partial mappings $\pi_0, \pi_1, \ldots$ with $\pi_i : \Delta_i \times W_i \to \Delta^{\mathcal{I}}$ and mappings $R_0, R_1, \ldots$ with $R_i : \mathsf{N_R} \times W_i \to \Delta_i \times \Delta_i$. We obtain our desired sets Δ, W, E in the limit. To start the construction of $\mathfrak{I}$, we set

- $\Delta_0 := \{d_0\}, \quad W_0 := \{w_0\}, \quad E_0 := \emptyset$
- $\pi_0(d_0, w) := d$ with d the root of $\mathcal{I}$
- $R_0(r, w_0) := \emptyset$ for all $r \in \mathsf{N_R}$

For the induction step, we start by setting $\Delta_i = \Delta_{i-1}, W_i = W_{i-1}, E_i = E_{i-1} R_i := R_{i-1}$ and $\pi_i = \pi_{i-1}$. Then, we proceed as follows:

(I) Let $d \in \Delta_i, w \in W_i$ such that $\pi_i(d, w) = e$ and $(e, f) \in r^{\mathcal{I}}$ for some $r \in \mathsf{N_R}$. Then, add a fresh element d' to Δ_i, and set $\pi_i(d', w) := f$.

(II) Let $d \in \Delta_i, w \in W_i$ such that $\pi_i(d, w) = d_A$ for some $A \in \mathsf{N_C} \cup \{\top\}$ and $\mathcal{T}' \models A \sqsubseteq \exists r.B$. Then, add a fresh element d' to Δ_i, add (d, d') to $R_i(r, w)$ and set $\pi_i(d', w) := d_B$;

(III) Let $d \in \Delta_i, w \in W_i$ such that $\pi_i(d, w) = e$ and $(e, f) \in \mathsf{succ}_\Diamond^{\mathcal{I}}$. Then, add fresh worlds w' and w'' to W_i, add (w, w') and (w', w'') to E_i and, set $\pi_i(d, w') := d_\top$; and $\pi_i(d, w'') := f$.

(IV) Let $d \in \Delta_i, w \in W_i$ such that $\pi_i(d, w) = d_A$ and $\mathcal{T}' \models A \sqsubseteq \exists \mathsf{succ}_\Diamond.B$. Then, add fresh worlds w' and w'' to W_i, add (w, w') and (w', w'') to E_i and, set $\pi_i(d, w') := d_\top$; and $\pi_i(d, w'') := d_B$.

(V) Let $d \in \Delta_i, w \in W_i$ such that $\pi_i(d, w) = e$ and $(e, f) \in \mathsf{succ}_\Box^{\mathcal{I}}$. Then, add a fresh world w' to W_i, add (w, w') to E_i and set $\pi_i(d, w') := f$.

(VI) Let $d \in \Delta_i, w \in W_i$ such that $\pi_i(d, w) = d_A$ for some $A \in \mathsf{N_C} \cup \{\top\}$, and $\mathcal{T}' \models A \sqsubseteq \exists \mathsf{succ}_\Box B$. Then, add a fresh world w' to W_i, add (w, w') to E_i and, set $\pi_i(d, w') := d_A$.

(VII) Let $d \in \Delta_i$ and $w_0 \ldots w_k \in W_i$ such that for all $0 \leq j < k$, $(w_j, w_{j+1}) \in E_i, \pi_i(d, w_j)$ is not defined and $\pi_i(d, w_k)$ defined. Then, for all $j < k$ set $\pi_i(d', w_j) := \pi_i(d', w_k)$.

(VIII) Let $d \in \Delta_i$ and $w, w' \in W_i$ such that $(w, w') \in E_i$, $\pi_i(d, w')$ is not defined and $\pi_i(d, w)$ defined. Then, set $\pi_i(d, w') := d_\top$.

Finally, set $\Delta := \bigcup_{i\geq 0} \Delta_i, W := \bigcup_{i\geq 0} W_i, E := \bigcup_{i\geq 0} E_i$. The temporal interpretation $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w\in W})$ is then given by:

$$\begin{array}{rcl} A^{\mathfrak{I},w} & = & \{d \in \Delta \mid \pi(d, w) \in A^{\mathcal{I}}\} \cup \\ & & \{d \in \Delta \mid \pi(d, w) = d_B \text{ and } \mathcal{T}' \models B \sqsubseteq A\} \\ & & \text{for all } A \in \mathsf{N_C} \setminus \{M_A\} \\ r^{\mathfrak{I},w} & = & \{(d, d')\Delta \times \Delta \mid (\pi(d, w), \pi(d', w)) \in r^{\mathcal{I}}\} \cup \\ & & \{(d, d') \in \Delta \times \Delta \mid (d, d') \in R(r, w)\} \text{ for all } r \in \mathsf{N_R}. \end{array}$$

Claim: For all $d, e \in \Delta^{\mathfrak{I}}$, $w \in W$ and basic concepts C we have:

1. If $\pi(d, w) = d_A$ then
   - $d \in C^{\mathfrak{I},w}$ iff $A \sqsubseteq_{\mathcal{T}'} \hat{C}$;
   - $(d, e) \in r^{\mathfrak{I},w}$ iff $A \sqsubseteq_{\mathcal{T}'} \exists r.B$. and $\pi(e, w) = d_B$
2. else
   - $d \in C^{\mathfrak{I},w}$ iff $\pi(d, w) \in \hat{C}^{\mathcal{I}}$;
   - $(d, e) \in r^{\mathfrak{I},w}$ iff $(\pi(d, w), \pi(e, w)) \in r^{\mathcal{I}}$.

Proof of Claim. We prove the statement by structural induction.

- Let $C = A \in \mathsf{N_C}$. Then it follows directly from the definitions.
- Let $C = \exists r.A$. Follows from definition of i
- Let $C = \mathbf{E}\Diamond A$, i.e., $\hat{C} = \exists \mathsf{succ}_\Diamond.A$

  1. $\Leftarrow$: We have that $\pi(d, w) = d_B$ and $\mathcal{T}' \models B \sqsubseteq \exists \mathsf{succ}_\Diamond.A$. By rule (IV), there exist w', w'' such that $\{(w, w'), (w'w'')\} \subseteq E$ and $\pi(d, w'') := d_A$. Now, by I.H., $d \in A^{\mathfrak{I},w''}$. Therefore, $d \in (\mathbf{E}\Diamond A)^{\mathfrak{I},w}$.

  $\Rightarrow$: We have that $\pi(d, w) = d_B$ and $d \in (\mathbf{E}\Diamond A)^{\mathfrak{I},w}$. Then by the semantics of $\mathbf{E}\Diamond A$, there exists a path $w_0 w_1 \ldots$ with $w = w_0$ such that $d \in A^{\mathfrak{I},w_j}$ for some $j \geq 0$. Note that by construction $\pi(d, w_j) = d_{B'}$ for some B', and by I.H., $\mathcal{T}' \models B' \sqsubseteq A$. In particular, we have that $\pi(d, w), \pi(d, w_1), \ldots \pi(d, w_j)$ was defined by the application of the rules (IV), (VI), or (VIII). Now, we distinguish the following cases:

  – $\pi(d, w_0), \pi(d, w_1), \ldots \pi(d, w_j)$ was defined by applying rule (IV), then we have that the following GCIs hold

  $$\begin{array}{rcl} \mathcal{T}' \models B & \sqsubseteq & \exists \mathsf{succ}_\Diamond.A_1, \\ \mathcal{T}' \models A_1 & \sqsubseteq & \exists \mathsf{succ}_\Diamond.A_2, \\ & \ldots & \\ \mathcal{T}' \models A_{j-1} & \sqsubseteq & \exists \mathsf{succ}_\Diamond.B' \end{array}$$

  Therefore, $\mathcal{T}' \models B \sqsubseteq \mathsf{succ}_\Diamond.A$. Since $\mathcal{T}' \models \mathsf{succ}_\Box \sqsubseteq \mathsf{succ}_\Diamond$ the same argument follows if we apply rule (VI), or both (IV) and (VI) to define $\pi(d, w_0), \pi(d, w_1), \ldots \pi(d, w_j)$.

  – A fragment $\pi(d, w_i) \ldots \pi(d, w_j)$, $0 \leq i \leq j$ was defined by rule (VIII). Then, $\pi(d, w_j) = d_\top$ and by I.H., $\mathcal{T}' \models \top \sqsubseteq A$. In particular, $\mathcal{T}' \models B \sqsubseteq A$, and since $\mathsf{succ}_\Diamond$ is reflexive $\mathcal{T}' \models B \sqsubseteq \exists \mathsf{succ}_\Diamond.A$.

2. $\Leftarrow$: Let $\pi(d,w) \in \hat{C}^{\mathcal{I}}$. Then, $\pi(d,w) \in (\exists \mathsf{succ}_{\Diamond}.A)^{\mathcal{I}}$. By the semantics, there is exists a $e \in \Delta^{\mathcal{I}}$ such that $e \in A^{\mathcal{I}}$ and $(\pi(d,w),e) \in \mathsf{succ}_{\Diamond}^{\mathcal{I}}$. By rule (II), there exist a $w', w'' \in W$ such that $\{(w,w'),(w',w'')\} \subseteq E$ and $\pi(d,w'') = e$. By I.H., $d \in A^{\mathfrak{I},w''}$. Therefore, $d \in (\mathbf{E}\Diamond A)^{\mathfrak{I},w}$.
$\Rightarrow$: Let $d \in (\mathbf{E}\Diamond A)^{\mathfrak{I},w}$. Then, by the semantics of $\mathbf{E}\Diamond A$, there exists a path $w_0w_1\dots$ with $w = w_0$ such that $d \in A^{\mathfrak{I},w_j}$ for some $j \geq 0$. We show that $(\pi(d,w_i),\pi(d,w_{i+1})) \in \mathsf{succ}_{\Diamond}^{\mathcal{I}}$ for all $i < j$, and then by transitivity of $\mathsf{succ}_{\Diamond}$, $(\pi(d,w),\pi(d,w_j)) \in \mathsf{succ}_{\Diamond}^{\mathcal{I}}$. Let i be arbitrary from $[0,\dots,j-1]$. First note that not both $\pi(d,w_i)$ and $\pi(d,w_{i+1})$ were defined by rule (I). Hence, we distinguish the following cases:

- $\pi(d,w_{i+1})$ was defined by rule (III), then, by definition, $(\pi(d,w_i),\pi(d,w_{i+1})) \in \mathsf{succ}_{\Diamond}^{\mathcal{I}}$
- $\pi(d,w_{i+1})$ was defined by rule (V), then $(\pi(d,w_i),\pi(d,w_{i+1})) \in \mathsf{succ}_{\Box}^{\mathcal{I}}$, and then, by $\mathsf{succ}_{\Box} \sqsubseteq \mathsf{succ}_{\Diamond}$, to $\mathsf{succ}_{\Diamond}^{\mathcal{I}}$.
- $\pi(d,w_{i+1})$ was defined by rule (VII), then $\pi(d,w_i) = \pi(d,w_{i+1}) = \pi(d,w_k)$ for some $i+1 < k \leq j$. Since, $\mathsf{succ}_{\Diamond}$ is reflexive, $(\pi(d,w_i),\pi(d,w_{i+1})) \in \mathsf{succ}_{\Diamond}^{\mathcal{I}}$.

Now, by I.H., $\pi(d,w_j) \in A^{\mathcal{I}}$, and together with $(\pi(d,w),\pi(d,w_j)) \in \mathsf{succ}_{\Diamond}^{\mathcal{I}}$ implies $\pi(d,w) \in (\exists \mathsf{succ}_{\Diamond}.A)^{\mathcal{I}}$. We can also have the case that $\pi(d,w_{i+1})$ was defined by rule (VIII). Then, $\pi(d,w_{i+1}) = d_{\top}$. By claim 1, $\top \sqsubseteq_{\mathcal{T}'} \exists \mathsf{succ}_{\Diamond}.A$. Therefore, $\pi(d,w) \in (\exists \mathsf{succ}_{\Diamond}.A)^{\mathcal{I}}$

- $C = \mathbf{E}\Box A$ i.e., $\hat{C} = M_A$
  1. $\Leftarrow$: We have that $\pi(d,w) = d_B$ and $\mathcal{T}' \models B \sqsubseteq M_A$. Since $\mathcal{T}' \models M_A \sqsubseteq A \sqcap \exists \mathsf{succ}_{\Box}.M_A$ (†). Then, $\mathcal{T}' \models B \sqsubseteq A \sqcap \exists \mathsf{succ}_{\Box}.M_A$. Thus, by rule (VI) there exist $w' \in W$ such that $(w,w') \in E$ and $\pi(d,w') = d_B$. We have that by (†), $B \sqsubseteq_{\mathcal{T}'} A$, and by I.H, $d \in A^{\mathfrak{I},w}$ and $d \in A^{\mathfrak{I},w'}$. Since $\mathsf{succ}_{\Box}$ is total and by (†) is not hard to see that we obtain an infinite sequence $w_0w_1w_2\dots$ with $w = w_0$ and $w_1 = w'$ such that for all $i \geq 0$, $d \in A^{\mathfrak{I},w_i}$. Therefore, $d \in (\mathbf{E}\Box A)^{\mathfrak{I},w}$.
  $\Rightarrow$: We have that $\pi(d,w) = d_B$ and $d \in (\mathbf{E}\Box A)^{\mathfrak{I},w}$. By the semantics of $\mathbf{E}\Box A$, there exists a path $w_0w_1\dots$ with $w = w_0$ such that for all $j \geq 0$ $d \in A^{\mathfrak{I},w_j}$. By (†), to show $\mathcal{T}' \models B \sqsubseteq A \sqcap \exists \mathsf{succ}_{\Box}.M_A$. In particular, we have that $\pi(d,w), \pi(d,w_1),\dots$ was defined by the application of the rules (IV), (VI), or (VIII). First, assume $w_0, w_1, \dots$ where introduced by rule (VI) to satisfy $\mathbf{E}\Box A$ i.e., $\pi(d,w) = d_B$ and $B \sqsubseteq_{\mathcal{T}'} \exists \mathsf{succ}_{\Box}.M_A$, then we are done. Otherwise, we distinguish the following cases:
  - The world w_1 was introduced by rule (IV) then $\pi(d,w_1)$ is defined by rule (VIII). Analogously $\pi(d,w_1)$ is defined by rule (VIII), if w_1 was introduced by rule (VI) for $B \sqsubseteq_{\mathcal{T}'} \exists \mathsf{succ}_{\Box}.B'$.
  - $\pi(d,w_1)$ is defined by rule (VIII). Then, $\pi(d,w_1) = d_{\top}$ and by I.H., $\mathcal{T}' \models \top \sqsubseteq M_A$. In particular, $\mathcal{T}' \models B \sqsubseteq M_A$. Therefore, $\mathcal{T}' \models B \sqsubseteq A \sqcap \exists \mathsf{succ}_{\Box}.M_A$.

  2.$\Leftarrow$: Let $\pi(d,w) \in \hat{C}^{\mathcal{I}}$. Then, $\pi(d,w) \in M_A^{\mathcal{I}}$. Since $\mathcal{T}' \models M_A \sqsubseteq A \sqcap \exists \mathsf{succ}_{\Box}.M_A$ (†), $\pi(d,w) \in (A \sqcap \exists \mathsf{succ}_{\Box}.M_A)^{\mathcal{I}}$. By the semantics, there is exists a $e \in \Delta^{\mathcal{I}}$ such that $e \in M_A^{\mathcal{I}}$ and $(\pi(d,w),e) \in \mathsf{succ}_{\Box}^{\mathcal{I}}$. By rule (V), there exists a $w' \in W$ such that $(w,w') \in E$ and $\pi(d,w') = e$. Then, by (†) and by I.H., $d \in A^{\mathfrak{I},w'}$. Using that $\mathsf{succ}_{\Box}$ is total and (†) one can prove by induction on the number of applications of (V) that we obtain an infinite path $w_0w_1w_2\dots \in W$ with $w_0 = w$ and $w_1 = w'$ such that for each $i \geq 0$, $d \in A^{\mathfrak{I},w_i}$. Therefore, $d \in (\mathbf{E}\Box A)^{\mathfrak{I},w}$.
  $\Rightarrow$: Let $d \in (\mathbf{E}\Box A)^{\mathfrak{I},w}$. Then, by the semantics of $\mathbf{E}\Box A$, there exists a path $w_0w_1\dots$ with $w = w_0$ such that for all $j \geq 0$ $d \in A^{\mathfrak{I},w_j}$. Since $\mathcal{T}' \models M_A \sqsubseteq A \sqcap \exists \mathsf{succ}_{\Box}.M_A$ (†), to show $(\pi(d,w_0),\pi(d,w_1)) \in \mathsf{succ}_{\Box}^{\mathcal{I}}$ and $\pi(d,w_1) \in M_A^{\mathcal{I}}$. Note that not both $\pi(d,w_0)$ and $\pi(d,w_1)$ were defined by rule (I). First, we assume that the worlds $w_0, w_1, \dots$ where introduced by repeated applications of rule (V). In this case, by definition $(\pi(d,w_0),\pi(d,w_1)) \in \mathsf{succ}_{\Box}^{\mathcal{I}}$, and by I.H. $\pi(d,w_1) \in M_A^{\mathcal{I}}$. Then, $\pi(d,w) \in (\exists \mathsf{succ}_{\Box}.M_A)^{\mathcal{I}}$. Otherwise, $\pi(d,w_1)$ was defined by the following rules:
  - If the world w_1 was introduce by rule (IV) then $\pi(d,w_1)$ is defined by rule (VIII).
  - $\pi(d,w_1)$ was defined by rule (VII), then $\pi(d,w_0) = \pi(d,w_1) = \pi(d,w_k)$ for some $k \geq 1$ but then $\pi(d,w_{k+1})$ is defined either by rule (V) or (VIII) falling in the other cases.
  - $\pi(d,w_1)$ was defined by rule (VIII), then $\pi(d,w_1) = d_{\top}$. Then, by I.H., $\top \sqsubseteq_{\mathcal{T}'} A$ and $\mathsf{succ}_{\Box}$ is total, $\top \sqsubseteq_{\mathcal{T}'} \exists \mathsf{succ}_{\Box}.M_A$. Therefore, $\pi(d,w_0) \in (\exists \mathsf{succ}_{\Box}.M_A)^{\mathcal{I}}$.

  Now we show that $\mathfrak{I} \models \mathcal{T}$. Let $X_1 \sqcap X_2 \sqcap \dots \sqcap X_n \sqsubseteq X \in \mathcal{T}$.
- Assume first $d \in X_i^{\mathfrak{I},w}$ for all $i \in 1,\dots,n$. By our claim, $d \in X_i^{\mathfrak{I},w}$. Since $\mathcal{I} \models \mathcal{T}'$ and $\hat{X}_1 \sqcap \dots \sqcap \hat{X}_n \sqsubseteq X \in \mathcal{T}$, we have also $\pi(d,w) \in \hat{X}^{\mathcal{I}}$, hence, again by the claim, $d \in X^{\mathfrak{I},w}$.
- Assume now $\pi(d,w) = d_A$ and $d \in X_i^{\mathfrak{I},w}$ for all $i \in 1,\dots,n$. By our claim, $A \sqsubseteq_{\mathcal{T}'} \hat{X}_i$. Since $X_1 \sqcap \dots \sqcap X_k \sqsubseteq X \in \mathcal{T}'$, we have also $A \sqsubseteq_{\mathcal{T}'} X$, thus again by our claim $d \in X^{\mathfrak{I},w}$.

Obviously $d \in A^{\mathfrak{I},w} \setminus B^{\mathfrak{I},w}$. Therefore, $A \sqsubseteq_{\mathcal{T}} B$.

$\Leftarrow$) We show the contrapositive, i.e., if $\mathcal{T} \models A \not\sqsubseteq B$ then $\mathcal{T}' \models A \not\sqsubseteq B$ for all concept names A, B. Let $\mathfrak{I} = (\Delta, T, \{\mathcal{I}_w\}_{w \in W})$ be a model of $\mathcal{T}$ such that $d_0 \in A^{\mathcal{I},\varepsilon} \setminus B^{\mathcal{I},\varepsilon}$ for some domain element d_0. We construct a model $\mathcal{J}$ of $\mathcal{T}'$ that has some $d \in A^{\mathcal{J}} \setminus B^{\mathcal{J}}$. Define $\Delta^{\mathcal{J}} := \Delta \cup \{d_A, e_A \mid A \in N_C\}$ and $\cdot^{\mathcal{J}}$ as follows:

$$
\begin{array}{rcl}
A^{\mathcal{J}} & = & A^{\mathfrak{I}} \cup \{d_B \mid \mathcal{T} \models B \sqsubseteq A\} \\
& \cup & \{e_B \mid \mathcal{T} \models \mathbf{E}\Box B \sqsubseteq A\} \\
M_A^{\mathcal{J}} & = & \{(w,d) \mid d \in (\mathbf{E}\Box A)^{\mathfrak{I},w}\} \\
& \cup & \{d_B \mid \mathcal{T} \models B \sqsubseteq \mathbf{E}\Box A\} \\
& \cup & \{e_B \mid \mathcal{T} \models \mathbf{E}\Box B \sqsubseteq \mathbf{E}\Box A\} \\
r^{\mathcal{J}} & = & r^{\mathfrak{I}} \\
& \cup & \{(d_A, d_B) \mid \mathcal{T} \models A \sqsubseteq \exists r.B\} \\
& \cup & \{(e_A, d_B) \mid \mathcal{T} \models \mathbf{E}\Box A \sqsubseteq \exists r.B\} \\
succ_{\Box}^{\mathcal{J}} & = & \{((w,d), e_A) \mid d \in (\mathbf{E}\Box A)^{\mathfrak{I},w}\} \\
& \cup & \{(d_A, e_B) \mid \mathcal{T} \models A \sqsubseteq \mathbf{E}\Box B\} \\
& \cup & \{(e_A, e_B) \mid \mathcal{T} \models \mathbf{E}\Box A \sqsubseteq \mathbf{E}\Box B\} \\
succ_{\Diamond}^{\mathcal{J}} & = & \mathsf{cl}^{rt}(E \\
& \cup & \{(d_A, e_B) \mid \mathcal{T} \models A \sqsubseteq \mathbf{E}\Diamond B\} \\
& \cup & \{(e_A, e_B) \mid \mathcal{T} \models \mathbf{E}\Box A \sqsubseteq \mathbf{E}\Diamond B\} \\
& \cup & \{((w,d), e_A) \mid d \in (\mathbf{E}\Diamond A)^{\mathfrak{I},w}\})
\end{array}
$$

where cl^{rt} is the transitive and reflexive closure of E.
Claim. For all basic concepts C we have

1. $(w,d) \in \hat{C}^{\mathcal{J}}$ iff $(w,d) \in C^{\mathfrak{I}}$;
2. $d_A \in \hat{C}^{\mathcal{J}}$ iff $A \sqsubseteq_{\mathcal{T}} C$;
3. $e_A \in \hat{C}^{\mathcal{J}}$ iff $\mathbf{E}\Box A \sqsubseteq_{\mathcal{T}} C$.

Proof of Claim. We prove the statement by structural induction.

- If $C = A$ is a concept name, it follows directly from our definitions.

- Let $\hat{C} = C = \exists r.D$ for some concept name D. Then both 1., 2., and 3. follow immediately from the construction.
- Let $C = \mathbf{E}\Diamond D$, i.e., $\hat{C} = \exists succ_\Diamond.D$.
$\Rightarrow$: Let first be $(w, d) \in \hat{C}^{\mathcal{J}}$. By definition of $succ_\Diamond^{\mathfrak{J}}$, there is either some $(w', d) \in D^{\mathcal{J}}$ such that $w <^{rt} w'$ in which case we are done, or there is a sequence $(w_0, d), (w_1, d), \ldots, (w_k, d), e_{A_1}, \ldots, e_{A_\ell}$ with $w_0 = w$, $w_i < w_{i+1}$, $(w_k, d) \in (\mathbf{E}\Box A_1)^{\mathfrak{J}}$, $A_i \sqsubseteq_{\mathcal{T}} \mathbf{E}\Box A_{i+1}$, and finally $A_\ell = D$. Then, by the semantics, $(w_k, d) \in (\mathbf{E}\Box A_\ell)^{\mathfrak{J}} \subseteq (\mathbf{E}\Diamond A_\ell)^{\mathfrak{J}}$. By temporal semantics, thus also $(w, d) \in (\mathbf{E}\Diamond D)^{\mathfrak{J}}$, since $A_\ell = D$.
Let now be $d_A \in \hat{C}^{\mathcal{J}}$. By the definition of $succ_\Diamond^{\mathcal{J}}$, there is some $e_B \in D^{\mathcal{J}}$ with $(d_A, e_B) \in succ_\Diamond^{\mathcal{J}}$, thus $B \sqsubseteq_{\mathcal{T}} D$ and $A \sqsubseteq_{\mathcal{T}} \mathbf{E}\Diamond B$. Therefore, $A \sqsubseteq_{\mathcal{T}} \mathbf{E}\Diamond D$.
Let now be $e_A \in \hat{C}^{\mathcal{J}}$. By the definition of $succ_\Diamond^{\mathcal{J}}$, there is some $e_B \in D^{\mathcal{J}}$ with $(d_A, e_B) \in succ_\Diamond^{\mathcal{J}}$, thus $B \sqsubseteq_{\mathcal{T}} D$ and $\mathbf{E}\Box A \sqsubseteq_{\mathcal{T}} \mathbf{E}\Diamond B$. Therefore, $\mathbf{E}\Box A \sqsubseteq_{\mathcal{T}} \mathbf{E}\Diamond D$.
$\Leftarrow$: Let first be $(w, d) \in (\mathbf{E}\Diamond D)^{\mathfrak{J}}$. Thus, there is a w' with $w <^{rt} w'$ and $(w', d) \in D^{\mathfrak{J}}$. By definition of the interpretation $\mathcal{J}$, $(w, d) \in (\exists succ_\Diamond.D)^{\mathcal{J}}$.
Let now be $A \sqsubseteq_{\mathcal{T}} \mathbf{E}\Diamond D$. By definition $(d_A, e_D) \in succ_\Diamond^{\mathcal{J}}$. Trivially, $e_D \in D^{\mathcal{J}}$, thus $d_A \in (\exists succ_\Diamond.D)^{\mathcal{J}}$.
Let now be $\mathbf{E}\Box A \sqsubseteq_{\mathcal{T}} \mathbf{E}\Diamond D$. By definition $(e_A, e_D) \in succ_\Diamond^{\mathcal{J}}$. Trivially, $e_D \in D^{\mathcal{J}}$, thus $d_A \in (\exists succ_\Diamond.D)^{\mathcal{J}}$.
- Let $C = \mathbf{E}\Box D$, i.e., $\hat{C} = M_D$.
The claim follows directly from the definition of $M_D^{\mathcal{J}}$

Now we show that $\mathcal{J} \models \mathcal{T}'$. First observe that (i) $succ_\Diamond^{\mathcal{J}}$ is reflexive and transitive, (ii) $succ_\Box^{\mathcal{J}} \subseteq succ_\Diamond^{\mathcal{I}}$, and (iii) $M_A \sqsubseteq A \sqcap \exists succ_\Box.M_A$ is satisfied:

- Assume first some $(w, d) \in M_A^{\mathcal{J}}$. By the claim, $d \in (\mathbf{E}\Box C)^{\mathcal{I},w}$. By the definition of $\mathcal{J}$ we have that $((w, d), e_A) \in succ_\Box^{\mathcal{J}}$ and $(e_A, e_A) \in succ_\Box^{\mathcal{J}}$. Obviously $e_A \in M_A^{\mathcal{J}}$ and $d \in C^{\mathfrak{J},w}$, thus $(w, d) \in A^{\mathcal{J}}$ by the claim. Together, this implies that $(w, d) \in (A \sqcap \exists \mathsf{succ}_\Box.M_A)^{\mathcal{J}}$.
- Assume now $d_B \in M_A^{\mathcal{J}}$. By the claim, $\mathcal{T} \models B \sqsubseteq \mathbf{E}\Box A$, hence $(d_B, e_A) \in succ_\Box^{\mathcal{J}}$ and $\mathcal{T} \models B \sqsubseteq A$, thus $d_B in A^{\mathcal{J}}$. Now, we can continue as in the first case.
- Assume now $e_B \in M_A^{\mathcal{J}}$. By the claim, $\mathcal{T} \models \mathbf{E}\Box B \sqsubseteq \mathbf{E}\Box A$, hence $(d_B, e_A) \in succ_\Box^{\mathcal{J}}$ and $\mathcal{T} \models B \sqsubseteq A$, thus $d_B \in A^{\mathcal{J}}$. Now, we can continue as in the first case.

It remains to show that the original GCIs are still satisfied. Let $\hat{X}_1 \sqcap \ldots \sqcap \hat{X}_k \sqsubseteq \hat{X} \in \mathcal{T}'$.

- Assume first $(w, d) \in \hat{X}_i^{\mathcal{J}}$ for all $i \in 1, \ldots, k$. By our claim, $d \in \hat{X}_i^{\mathfrak{J},w}$. Since $\mathfrak{J} \models \mathcal{T}$ and $X_1 \sqcap \ldots \sqcap X_k \sqsubseteq X \in \mathcal{T}$, we have also $d \in X^{\mathfrak{J},w}$, hence, again by the claim, $(w, d) \in \hat{X}^{\mathcal{J}}$.
- Assume now $d_A \in \hat{X}_i^{\mathcal{J}}$ for all $i \in 1, \ldots, k$. By our claim, $\mathcal{T} \models A \sqsubseteq X_i$. Since $X_1 \sqcap \ldots \sqcap X_k \sqsubseteq X \in \mathcal{T}$, we have also $\mathcal{T} \models A \sqsubseteq X$, thus $d_A \in \hat{X}^{\mathcal{J}}$.
- Assume finally $e_A \in \hat{X}_i^{\mathcal{J}}$ for all $i \in 1, \ldots, k$. By our claim, $\mathcal{T} \models \mathbf{E}\Box A \sqsubseteq X_i$. Since $X_1 \sqcap \ldots \sqcap X_k \sqsubseteq X \in \mathcal{T}$, we have also $\mathcal{T} \models \mathbf{E}\Box A \sqsubseteq X$, thus $e_A \in \hat{X}^{\mathcal{J}}$.

Obviously, $(w_0, d_0) \in A^{\mathcal{J}} \setminus B^{\mathcal{J}}$, thus $A \not\sqsubseteq_{\mathcal{T}'} B$. This finishes the proof of this direction.

❏

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

Proof. The proof is by reduction of the validity of QBFs, i.e., formulas of the form $\Psi = Q_1 x_1 \ldots Q_n x_n.\varphi$ where $Q_i \in \{\exists, \forall\}$ for $1 \leq i \leq n$ and φ a propositional formula in negation normal form. For a formal definition of QBFs, and their validity problem, please consult [19]. Given a QBF $\Psi = Q_1 x_1 \ldots Q_n x_n.\varphi$ with φ we construct in polynomial time a CTL$_{\mathcal{EL}}^{\mathbf{E}\Box,\mathbf{E}\Diamond,\mathbf{A}\Box}$ TBox $\mathcal{T}_\Psi$ such that for certain concept names A_0, B_0, we have $\mathcal{T}_\Psi \models A_0 \sqsubseteq B_0$ iff Ψ is valid.

- We introduce fresh concept names X_ψ for every subformula ψ of φ, so in particular for each variable x_i concept names X_{x_i} and $X_{\neg x_i}$.
- We construct a binary tree of depth n rooted in A_0 representing all possible evaluations of $\{x_1, \ldots, x_n\}$. and we use concept names A_i, $0 \leq i \leq n$ to distinguish the levels of the tree. Specifically, $\mathcal{T}_\psi$ contains the following CI for every $1 \leq i < n$:
$$A_i \sqsubseteq \mathbf{E}\Diamond(A_{i+1} \sqcap X_{x_{i+1}}) \sqcap \mathbf{E}\Diamond(A_{i+1} \sqcap X_{\neg x_{i+1}})$$
- Once we have decided the truth value of a variable we keep it through all descendants.
$$\begin{array}{rcl} X_{x_i} & \sqsubseteq & \mathbf{A}\Box X_{x_i} \text{ for all } 0 \leq i \leq n \\ X_{\neg x_i} & \sqsubseteq & \mathbf{A}\Box X_{\neg x_i} \text{ for all } 0 \leq i \leq n \end{array}$$
- We need to evaluate the formula in each leaf.
$A_n \sqcap X_\chi \sqcap X_\theta \sqsubseteq X_{\chi \wedge \theta}$ for all subformulas $\chi \wedge \theta$ of Ψ
$A_n \sqcap X_\chi$ and $A_n \sqcap X_\theta \sqsubseteq X_{\chi \vee \theta}$ for all subformulas $\chi \vee \theta$ of Ψ
- To evaluate the formula we proceed from the leafs to the root. We identify each level with a quantifier in Ψ. For every $1 \leq i \leq n$ with $Q_i = \exists$ we have
$$\begin{array}{rcl} A_{i-1} \sqcap \mathbf{E}\Diamond(A_i \sqcap X_{x_i} \sqcap X_\varphi) & \sqsubseteq & X_\varphi \\ A_{i-1} \sqcap \mathbf{E}\Diamond(A_i \sqcap X_{\neg x_i} \sqcap X_\varphi) & \sqsubseteq & X_\varphi \end{array}$$
while for $Q_i = \forall$ we have that
$$\begin{array}{rl} A_{i-1} & \sqcap \; \mathbf{E}\Diamond(A_i \sqcap X_{x_i} \sqcap X_\varphi) \\ & \sqcap \; \mathbf{E}\Diamond(A_i \sqcap X_{\neg x_i} \sqcap X_\varphi) \sqsubseteq X_\varphi \end{array}$$

It is not hard to check that Ψ is valid iff $\mathcal{T}_\Psi \models A_0 \sqsubseteq X_\varphi$. ❏

Theorem 7 *We consider the logic* $\mathcal{EL}^{\mathbf{E}\Diamond,\mathbf{A}\Diamond}$, *and set* $\mathcal{T} = \emptyset$ *and*

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

Proof. Now, we show that the above witnesses non-convexity.

Lemma 11 $\mathcal{T} \models C \sqsubseteq \bigsqcup D_i$ *but* $\mathcal{T} \not\models D_i$ *for* $0 < i \leq 2$.

Proof. For the former, let $\mathfrak{J}$ a model with $d \in C^{\mathfrak{J},\varepsilon}$. Then, $\exists j \geq 0, k \geq 0.(d \in A^{\mathfrak{J},\pi[j])} \wedge d \in B^{\mathfrak{J},\pi[k]})$ for all $\pi \in \mathsf{Paths}(\varepsilon)$. Then, for some $\pi' \in \mathsf{Paths}(\varepsilon)$ depending on whether $k \leq j$ or $j \leq k$ implies $d \in D_i^{\mathfrak{J},\varepsilon}$.

For the latter, we construct a temporal model $\mathfrak{J} = (\Delta, T, \{\mathcal{I}_w\}_{w \in W})$ with $\Delta = \{d\}$ and T a 1-ary tree such that $d \in C^{\mathcal{I},\varepsilon}$ and $d \notin D_1^{\mathcal{I},\varepsilon}$. Let $w_1 = \varepsilon \cdot 1$ and $w_2 = w_1 \cdot 1$. Then, set

$$\begin{array}{rcl} A^{\mathfrak{J},w_1} & := & \Delta \\ A^{\mathfrak{J},w} & := & \emptyset \text{ for } w \neq w_1 \\ B^{\mathfrak{J},w_2} & := & \Delta \\ B^{\mathfrak{J},w} & := & \emptyset \text{ for } w \neq w_2 \end{array}$$

It is clear $d \in (\mathbf{A}\Diamond A \sqcap \mathbf{A}\Diamond B)^{\mathfrak{J},\varepsilon}$ but $d \notin (\mathbf{E}\Diamond(A \sqcap \mathbf{E}\Diamond B))^{\mathfrak{J},\varepsilon}$. Following the previous ideas we can construct a model $\mathfrak{J}'$ such that $d \in C^{\mathfrak{J}',\varepsilon}$ but $d \notin D_2^{\mathfrak{J}',\varepsilon}$. ❏

Now that we have established the non-convexity of $\mathrm{CTL}_{\mathcal{EL}}^{\mathbf{A}\Diamond,\mathbf{E}\Diamond}$ standard proof techniques from [6] can be used to show EXPTIME-hardness. ❑

For the remaining BTLS consider in Theorem 7 non-convexity can be shown following the ideas of Lemma 11 . In particular, for all cases we use the TBox $\mathcal{T} = \emptyset$. For $\mathcal{EL}^{\mathbf{E}\Diamond,\mathbf{E}\bigcirc}$ set

$$\begin{array}{lcl} C & = & \mathbf{E}\Diamond A \\ D_1 & = & A \\ D_2 & = & \mathbf{E}\bigcirc\, \mathbf{E}\Diamond A \end{array}$$

For $\mathcal{EL}^{\mathbf{E}\mathcal{U}}$ set

$$\begin{array}{lcl} C & = & \mathbf{E}(A\mathcal{U}B) \\ D_1 & = & B \\ D_2 & = & A \sqcap \mathbf{E}(A\mathcal{U}B) \end{array}$$

Analogously, for the extensions with for all path quantifier $\mathbf{A}$.