Combining DL-Lite$^{\mathcal{N}}_{bool}$ with Branching Time: A gentle Marriage*

Víctor Gutiérrez-Basulto
Cardiff University, UK
gutierrezbasultov@cardiff.ac.uk

Jean Christoph Jung
Universität Bremen, Germany
jeanjung@informatik.uni-bremen.de

Abstract

We study combinations of the description logic DL-Lite$^{\mathcal{N}}_{bool}$ with the branching temporal logics CTL* and CTL. We analyse two types of combinations, both with rigid roles: (i) temporal operators are applied to concepts and ABox assertions, and (ii) temporal operators are applied to concepts and Boolean combinations of concept inclusions and ABox assertions. For the resulting logics, we present algorithms for the satisfiability problem and (mostly tight) complexity bounds ranging from EXPTIME to 3EXPTIME.

1 Introduction

Over the last 25 years, plenty of extensions of classical description logics (DLs) with an explicit temporal component have been investigated. The study of temporal DLs (TDLs) is motivated by the fact that, arguably, in almost every domain where ontologies are used many terms are described and classified based on certain temporal patterns. For instance, in the biomedical domain, where DL-ontologies are commonly used, diseases or findings are defined according to the evolution of specific symptoms or the repetition of certain patterns over time [Shankar *et al.*, 2008; O'Connor *et al.*, 2009; Crowe and Tao, 2015]. Another prominent application of TDLs is the representation of and reasoning about temporal conceptual data models (TCMs) [Artale *et al.*, 2007b; 2011; 2014] introduced in the context of temporal databases as extensions of classical conceptual models such as ER and UML.

The most popular approach to TDLs is to combine DLs with traditional temporal logics such as LTL or CTL* and provide a two-dimensional semantics, one dimension for time and the other for DL quantification, in the style of many-dimensional modal logics [Gabbay *et al.*, 2003]. In the construction of this kind of TDLs there are a number of design choices, depending on the desired level of interaction between the component logics. For example, we can choose whether temporal operators are applied to concepts, roles or TBox and ABox axioms. Unfortunately, many TDLs become undecidable if they allow to reason about the temporal evolution of both roles and concepts [Artale *et al.*, 2007a; Lutz *et al.*, 2008; Gutiérrez-Basulto *et al.*, 2014]. In fact, these undecidability results hold already if roles are declared *rigid* (not changing their interpretation over time), the temporal operators available are heavily restricted and the lightweight sub-Boolean DL $\mathcal{EL}$ is used. On the other hand, in their seminal work, Artale *et al.* [2007a] showed that TDLs based on LTL and members of the family of lightweight DLs *DL-Lite* [Artale *et al.*, 2009] can support rigid roles and temporal concepts without compromising decidability. Furthermore, it was recently shown [Artale *et al.*, 2014] that this type of TDLs, besides being complexity-wise well-behaved, are well-suited to encode many important aspects of TCMs.

The purpose of this paper is to make further progress in the study of temporal extensions of *DL-Lite*. We are particularly interested in setting the basis for the development of a fine-grained analysis of TDLs based on *DL-Lite* and the branching temporal logics CTL* and CTL. To this aim, we investigate combinations of the expressive member *DL-Lite*$^{\mathcal{N}}_{bool}$ of *DL-Lite* with CTL* and CTL, and provide algorithms for the satisfiability problem and (mostly tight) complexity bounds. We believe the obtained results are important because the understanding of this kind of TDLs was very limited; indeed, only initial results were available [Gutiérrez-Basulto *et al.*, 2014]. Moreover, in the light of the results by Artale *et al.* [2014] above, these TDLs might be well-suited to encode TCMs incorporating branching time to capture various *versions* of the different components of conceptual schemas over time [Golendziner and dos Santos, 1995; Moro *et al.*, 2001; de Matos Galante *et al.*, 2005], or *version* document models, also requiring branching time [Noronha *et al.*, 1998; Weitl *et al.*, 2009].

We look at two types of combinations: (i) CTL*-*Lite*$^{\mathcal{N}}_{bool}$ and CTL-*Lite*$^{\mathcal{N}}_{bool}$ with rigid roles in the case where temporal operators are applied to concepts and temporal ABox assertions of certain shape are allowed, too. (ii) we look again at the TDLs in (i), but additionally temporal operators can be applied to Boolean combinations of concept inclusions and ABox assertions. For both types of combinations, we develop a uniform algorithm (working for both CTL and CTL*-based TDLs) for satisfiability based on a combination of type-based abstractions and tree automata. This approach to satisfiability was originally introduced in [Gutiérrez-Basulto *et al.*, 2012] for combinations of CTL and CTL* with $\mathcal{ALC}$; here, we extend and adapt it to deal with ABoxes, unqualified number

*Gutiérrez-Basulto was funded by the EU's Horizon 2020 programme under the Marie Skłodowska-Curie grant No 663830.

restrictions and rigid roles. Note that, as discussed, if rigid roles are allowed in *qualified* existential restrictions, already combinations of sub-fragments of CTL with the sub-logic $\mathcal{EL}$ of $\mathcal{ALC}$ are undecidable [Gutiérrez-Basulto *et al.*, 2014].

Our results are as follows: For combinations of type (i), we develop a uniform algorithm for *knowledge base satisfiability* based on a combination of type-elimination with automata-theoretic approaches to temporal reasoning in CTL and CTL*, yielding tight 2EXPTIME and EXPTIME upper bounds for CTL*-*Lite*$^{\mathcal{N}}_{bool}$ and CTL-*Lite*$^{\mathcal{N}}_{bool}$, respectively. For combinations of type (ii), we devise a uniform approach to *formula satisfiability* that combines type-based abstractions, nondeterministic automata for CTL and CTL*, and two-way alternating tree automata. We obtain 3EXPTIME and 2EXPTIME upper bounds for CTL*-*Lite*$^{\mathcal{N}}_{bool}$- and CTL-*Lite*$^{\mathcal{N}}_{bool}$ formulas, respectively. For CTL-*Lite*$^{\mathcal{N}}_{bool}$, we get a matching lower bound.

We note in passing that previously developed techniques for combinations of LTL with *DL-Lite*$^{\mathcal{N}}_{bool}$ cannot be straightforwardly adapted to the branching case. For combinations of type (i), upper bounds cannot be obtained by adapting the two-step technique developed by Artale *et al.* [2014] because the second step of such technique relies on the past being unbounded, which is not the standard semantics for CTL and CTL*. For combinations of type (ii), elementary upper bounds cannot be derived by providing a satisfiability preserving translation into (a decidable sub-fragment of) the one-variable fragment of first-order branching temporal logic [Hodkinson *et al.*, 2002] because (to our knowledge) no elementary complexity bounds are known. In contrast, for LTL-*Lite*$^{\mathcal{N}}_{bool}$ formulas tight upper bounds are obtained via a satisfiability preserving translation into the one-variable fragment of first-order temporal logic [Artale *et al.*, 2007a].

An extended version with an appendix can be found in http://tinyurl.com/ktkgwqg.

2 Preliminaries

Syntax. We introduce the TDLs CTL*-*Lite*$^{\mathcal{N}}_{bool}$ and CTL-*Lite*$^{\mathcal{N}}_{bool}$ based on the DL *DL-Lite*$^{\mathcal{N}}_{bool}$. We consider a vocabulary of countably infinite disjoint sets of *concept names* $\mathsf{N_C}$, *role names* $\mathsf{N_R}$ and *individual names* $\mathsf{N_I}$, and assume that $\mathsf{N_R}$ is partitioned into two countably infinite sets of *rigid role names* $\mathsf{N_{rig}}$ and *local role names* $\mathsf{N_{loc}}$. CTL*-*Lite*$^{\mathcal{N}}_{bool}$*-state concepts* C, D and CTL*-*Lite*$^{\mathcal{N}}_{bool}$*-path concepts* $\mathfrak{C}, \mathfrak{D}$ are defined by the following grammar:

$$C, D \quad ::= \quad A \mid \geq n\, r \mid \neg C \mid C \sqcap D \mid \mathbf{E}\mathfrak{C}$$
$$\mathfrak{C}, \mathfrak{D} \quad ::= \quad C \mid \mathfrak{C} \sqcap \mathfrak{D} \mid \neg \mathfrak{C} \mid \bigcirc \mathfrak{C} \mid \Box \mathfrak{C} \mid \mathfrak{C}\, \mathcal{U}\, \mathfrak{D}$$

with $A \in \mathsf{N_R}$, $r \in \{s, s^- \mid s \in \mathsf{N_R}\}$ a *role*, C, D state concepts, $\mathfrak{C}, \mathfrak{D}$ path concepts and n a positive integer given in binary. *CTL-Lite*$^{\mathcal{N}}_{bool}$ is the fragment of CTL*-*Lite*$^{\mathcal{N}}_{bool}$ in which *temporal operators* $\bigcirc, \Box, \mathcal{U}$ must be immediately preceded by the *path quantifier* $\mathbf{E}$. From now on, the term *concept* refers to a state concept.

Roles and concepts of the form r^- and $\geq n\, r$ are called *inverse roles* and *number restrictions*, respectively. We identify r^- with $s \in \mathsf{N_R}$ if $r = s^-$, use standard abbreviations $\top, \bot, C \sqcup D, C \rightarrow D, \exists r$ and $\leq n\, r$, and temporal abbreviations $\mathbf{A}\mathfrak{C} = \neg\mathbf{E}\neg\mathfrak{C}$ and $\Diamond\mathfrak{C} = \neg\Box\neg\mathfrak{C}$. In CTL-*Lite*$^{\mathcal{N}}_{bool}$ the abbreviations $\mathbf{A}\bigcirc C$, $\mathbf{A}C\mathcal{U}D$, $\mathbf{E}\Diamond C$ and $\mathbf{A}\Diamond C$ are defined as in CTL [Clarke and Emerson, 1981] - cf. appendix.

A CTL*-*Lite*$^{\mathcal{N}}_{bool}$ *TBox* $\mathcal{T}$ is a finite set of *concept inclusions (CIs)* $C \sqsubseteq D$ with C, D CTL*-*Lite*$^{\mathcal{N}}_{bool}$ concepts[1] - CTL-*Lite*$^{\mathcal{N}}_{bool}$ *TBoxes* are defined analogously. An *ABox* $\mathcal{A}$ is a finite set of *concept assertions* and *role assertions* of the form

$$(\mathbf{P}\bigcirc)^i A(a),\ (\mathbf{P}\bigcirc)^i \neg A(a),\ (\mathbf{P}\bigcirc)^i s(a,b),\ (\mathbf{P}\bigcirc)^i \neg s(a,b),$$

where $\mathbf{P} \in \{\mathbf{A}, \mathbf{E}\}$, $A \in \mathsf{N_C}$, $s \in \mathsf{N_R}$, $\{a, b\} \subseteq \mathsf{N_I}$ and $(\mathbf{P}\bigcirc)^i$ denotes $\mathbf{P}\bigcirc$ i times, with i given in unary. To avoid clunky notation, from here on, we write $\mathbf{P}\bigcirc^i$ instead of $(\mathbf{P}\bigcirc)^i$. Moreover, we assume wlog. that $\mathbf{E}\bigcirc^i s(a,b) \notin \mathcal{A}$ if $\mathbf{A}\bigcirc^i s(a,b) \in \mathcal{A}$. A *knowledge base (KB)* $\mathcal{K}$ is a pair $(\mathcal{T}, \mathcal{A})$.

We use $\mathsf{ind}(\mathcal{A})$, $\mathsf{CN}(\mathcal{K})$, $\mathsf{Rol}(\mathcal{K})$ to denote, respectively, the set of all (i) individual names occurring in $\mathcal{A}$, (ii) concept names occurring in $\mathcal{K}$, (iii) role names occurring in $\mathcal{K}$ and their inverses; and $\mathsf{Rol}_{\mathsf{rig/loc}}(\mathcal{K})$, $\sharp^r_{\mathcal{T}}$ and $\sharp_{\mathcal{A}}$ to denote, respectively, (iv) the subset of rigid/local roles of $\mathsf{Rol}(\mathcal{K})$, (v) the set containing 1 and all numbers n such that $\geq n\, r$ occurs in $\mathcal{T}$ and (vi) all numbers $0 \leq n \leq i$ such that $\bigcirc^i\alpha$ occurs in $\mathcal{A}$.

Semantics. An *interpretation* $\mathfrak{I}$ based on an infinite tree $T = (W, E)$ is a structure $(\Delta^{\mathfrak{I}}, (\mathcal{I}_w)_{w \in W})$, where each $\mathcal{I}_w$ is a classical DL interpretation with domain $\Delta^{\mathfrak{I}}$, that is, for each $w \in W$ we have $a^{\mathcal{I}_w} \in \Delta^{\mathfrak{I}}$, $A^{\mathcal{I}_w} \subseteq \Delta^{\mathfrak{I}}$, and $r^{\mathcal{I}_w} \subseteq \Delta^{\mathfrak{I}} \times \Delta^{\mathfrak{I}}$. We additionally require that the interpretations of a rigid role name is the same at all $w \in W$. Moreover, we make the two common assumptions *constant domain assumption (CDA)*, that is, all $w \in W$ share the same domain $\Delta^{\mathfrak{I}}$, and *standard name assumption*, that is, we assume $a^{\mathfrak{I}_w} = a$ for all $a \in \mathsf{ind}(\mathcal{A})$, $w \in W$. From here on, we usually write $A^{\mathfrak{I},w}$ instead of $A^{\mathcal{I}_w}$, and refer to nodes in T as *time points* or *worlds*.

For a *path* $\pi = w_0 w_1 w_2 \cdots$ in T, we write $\pi[i]$ for w_i, $\pi[i..]$ for the path $w_i w_{i+1} \cdots$ and use $\mathsf{Paths}(w)$ to denote the set of all paths starting at node w. The mapping $\cdot^{\mathfrak{I},w}$ is then extended from concept names and role names as follows:

$$(\neg C)^{\mathfrak{I},w} = \Delta^{\mathfrak{I}} \setminus C^{\mathfrak{I},w},$$
$$(C \sqcap D)^{\mathfrak{I},w} = C^{\mathfrak{I},w} \cap D^{\mathfrak{I},w},$$
$$(r^-)^{\mathfrak{I},w} = \{(d', d) \mid (d, d') \in r^{\mathfrak{I},w}\},$$
$$(\geq n\, r)^{\mathfrak{I},w} = \{d \mid \sharp\{d' \mid (d, d') \in r^{\mathfrak{I},w}\} \geq n\},$$
$$(\mathbf{E}\mathfrak{C})^{\mathfrak{I},w} = \{d \in \Delta^{\mathfrak{I}} \mid d \in \mathfrak{C}^{\mathfrak{I},\pi} \text{ for some } \pi \in \mathsf{Paths}(w)\},$$

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

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

[1] Inclusions between path concepts are not admitted since they lead to undecidability [Hodkinson *et al.*, 2002; Lutz *et al.*, 2008].

The satisfaction relation $\models$ is defined as follows, where ε is the root world and $*$ stands for *all* if $\mathbf{P} = \mathbf{A}$ and for *some* if $\mathbf{P} = \mathbf{E}$.

$$\begin{array}{lll}
\mathfrak{I} \models C \sqsubseteq D & \text{iff} & C^{\mathfrak{I},w} \subseteq D^{\mathfrak{I},w}, \text{ for all } w \in W; \\
\mathfrak{I} \models \mathbf{P}\bigcirc^i A(a) & \text{iff} & a \in A^{\mathfrak{I},\pi[i]}, \text{ for } * \pi \in \mathsf{Paths}(\varepsilon); \\
\mathfrak{I} \models \mathbf{P}\bigcirc^i \neg A(a) & \text{iff} & a \notin A^{\mathfrak{I},\pi[i]}, \text{ for } * \pi \in \mathsf{Paths}(\varepsilon); \\
\mathfrak{I} \models \mathbf{P}\bigcirc^i s(a,b) & \text{iff} & (a,b) \in s^{\mathfrak{I},\pi[i]}, \text{ for } * \pi \in \mathsf{Paths}(\varepsilon); \\
\mathfrak{I} \models \mathbf{P}\bigcirc^i \neg s(a,b) & \text{iff} & (a,b) \notin s^{\mathfrak{I},\pi[i]}, \text{ for } * \pi \in \mathsf{Paths}(\varepsilon).
\end{array}$$

An interpretation $\mathfrak{I}$ is a *model of a TBox* $\mathcal{T}$, written $\mathfrak{I} \models \mathcal{T}$, if $\mathfrak{I} \models \alpha$ for all $\alpha \in \mathcal{T}$, and it is a *model of an ABox* $\mathcal{A}$, written $\mathfrak{I} \models \mathcal{A}$, if $\mathfrak{I} \models \alpha$ for all $\alpha \in \mathcal{A}$. Thus, a TBox $\mathcal{T}$ is interpreted globally and ABox assertions are interpreted with respect to the root world ε. Finally, $\mathfrak{I}$ is a *model of a KB* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, denoted by $\mathfrak{I} \models \mathcal{K}$, if $\mathfrak{I} \models \mathcal{T}$ and $\mathfrak{I} \models \mathcal{A}$.

Fragments. We consider the sub-language CTL-*Lite*$_{horn}$ of CTL-*Lite*$^{\mathcal{N}}_{bool}$ that disallows the constructor $\neg$ (and thus abbreviations $C \sqcup D$, etc) and numbers $n \geq 2$ in number restrictions $\geq n\, r$. In this context, we see the operator $\mathbf{A}\bigcirc$ as a primitive instead of as an abbreviation.

Reasoning Problem. We are interested in the *knowledge base satisfiability problem*: given a KB $\mathcal{K}$, determine whether there exists an interpretation $\mathfrak{I}$ such that $\mathfrak{I} \models \mathcal{K}$.

3 KB Satisfiability in CTL- & CTL*-*Lite*$^{\mathcal{N}}_{bool}$

We next devise an algorithm for the KB satisfiability problem in CTL- & CTL*-*Lite*$^{\mathcal{N}}_{bool}$, yielding tight ExpTime and 2ExpTime upper bounds, respectively. The lower bounds are inherited from CTL and CTL* [Fischer and Ladner, 1979; Vardi and Stockmeyer, 1985]. We present for both TDLs a uniform approach to satisfiability that amalgamates Pratt-style type elimination [Pratt, 1979] with automata-based techniques for temporal logics [Kupferman and Vardi, 2005; Vardi, 2006]. We particularly use the fact that for deciding whether a (propositional) CTL or CTL* formula φ is satisfiable, one can construct a *nondeterministic Büchi tree automaton (NBTA)* $\mathfrak{A}_\varphi$ that accepts all (tree) models of φ, and then check whether any tree is accepted by $\mathfrak{A}_\varphi$, see appendix for details. The overall approach is sanctioned by the rather limited interaction between the temporal and DL dimensions, which allows us to 'independently' reason about the temporal evolution of each domain element, and then use all such one-dimensional temporal models to construct a single two-dimensional one.

In the rest of the paper, we use standard terminology for trees [Vardi, 1998]. We will say that a tree T is *k-ary*, $k \geq 1$, if every node of T has exactly k successors. Let Σ be a finite alphabet. A Σ*-labelled tree* is a pair (T, τ) with T a tree and $\tau : W \to \Sigma$ assigns a letter from Σ to each node.

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be the KB whose satisfiability is to be decided, with $\mathcal{T}$ formulated in CTL-*Lite*$^{\mathcal{N}}_{bool}$ or CTL*-*Lite*$^{\mathcal{N}}_{bool}$. We assume wlog. that $\mathcal{T}$ is of the form $\{\top \sqsubseteq C_\mathcal{T}\}$ and that if $\mathbf{P}\bigcirc^i s(a,b) \in \mathcal{A}$ then $\mathbf{P}\bigcirc^i s^-(b,a) \in \mathcal{A}$, and use $\mathsf{ccl}(\mathcal{K})$ to denote the *concept closure* under subconcepts and single negation of $\{C_\mathcal{T}\} \cup \{C \mid C(a) \in \mathcal{A}\} \cup \{\exists r \mid r \in \mathsf{Rol}(\mathcal{K})\} \cup \{\mathbf{P}\bigcirc^i \geq n\, r \mid \mathbf{P}\bigcirc^i r(a,b) \in \mathcal{A}_{\mathsf{rol}} \wedge n \in \sharp^r_\mathcal{T}\}$ where $\mathcal{A}_{\mathsf{rol}} = \{\mathbf{P}\bigcirc^j R(a,b) \mid \mathbf{P}\bigcirc^i R(a,b) \in \mathcal{A}, j \leq i\}$ with R of the form r or $\neg r$.

A *concept type for* $\mathcal{K}$ is a set $t \subseteq \mathsf{ccl}(\mathcal{K})$ such that (i) $C_\mathcal{T} \in t$ and (ii) if $\geq n\, r \in t$, then $\geq m\, r \in t$, for all $m \in \sharp^r_\mathcal{T}$ with $m < n$; and $\mathsf{tp}(\mathcal{K})$ denotes the set of all concept types for $\mathcal{K}$. An *ABox type for* $\mathcal{K}$ is a pair (t, a) with $t \subseteq \mathsf{ccl}(\mathcal{K}) \cup \mathcal{A}_{\mathsf{rol}}$ and $a \in \mathsf{ind}(\mathcal{A})$ such that t satisfies (i)-(ii) above and (iii) if $\mathbf{P}\bigcirc^i R(b,c) \in t$, then $b = a$ or $c = a$; and $\mathsf{atp}(\mathcal{K})$ denotes the set of all ABox types for $\mathcal{K}$. From here on, we write t_a instead of (t, a) and use $\alpha \in t_a$ to denote that $\alpha \in t$. We say *type for* $\mathcal{K}$ to refer to either a concept or ABox type for $\mathcal{K}$, and freely use $\geq 1\, r$ or $\exists r$. If no confusion arises, we omit the name a in t_a and write t.

We now introduce the temporal component in our type-based abstraction. A *temporal type for* $\mathcal{K}$ is a pair (t, i) where $t \in \mathsf{tp}(\mathcal{K}) \cup \mathsf{atp}(\mathcal{K})$ is a type for $\mathcal{K}$ and $i \geq 0$ denotes the distance of a world w from the root time point. For any $n_0 \geq 0$, $\mathsf{ttp}_{n_0}(\mathcal{K})$ denotes the set of all temporal types (t, i) for $\mathcal{K}$ with $i \leq n_0$.

As the next step, in order to use the automata machinery for temporal reasoning, we surrogate the 'DL-component' from our logics. We start by introducing a surrogation on types that will allow us to use them as an alphabet of models of the (propositional) temporal formulas defined below. For a type t, let $\bar{t}$ denote the result of replacing (i) every $C \in t$, $C \notin \mathsf{CN}(\mathcal{K})$, with a fresh propositional variable X_C, and (ii) every $\beta = \mathbf{P}\bigcirc^i R(a,b) \in t$ with a fresh propositional variable X_β. Let cn be the set of resulting propositional variables, including $\mathsf{CN}(\mathcal{K})$. As the final step, we surrogate concepts and role assertions so as to obtain temporal formulas. For $C \in \mathsf{ccl}(\mathcal{K})$, $\bar{C}$ denotes the result of replacing in C every subconcept $\geq n\, r$ with X^n_r and $\sqcap$ with $\wedge$; for $\beta = \mathbf{P}\bigcirc^i R(a,b) \in \mathcal{A}_{\mathsf{rol}}$, $\bar{\beta}$ denotes the result of replacing in β $R(a,b)$ with $X_{R(a,b)}$.

We are now ready to describe the overall decision procedure, we will establish later on an appropriate n_0 bound. Our algorithm performs *type-elimination*, similar to what has been done for combinations (without rigid roles and ABoxes) of $\mathcal{ALC}$ with CTL and CTL* [Gutiérrez-Basulto *et al.*, 2012]. The algorithm starts with the set $S_0 = \mathsf{ttp}_{n_0}(\mathcal{K})$ and obtains the set S_{j+1} from S_j by removing temporal types that, intuitively, cannot appear in any model of $\mathcal{K}$: A temporal type (t_a, i) is eliminated from S_j if it is *not realisable* in S_j. Abusing notation, in the next definition t_a denotes a concept or ABox type; in the former case, the subindex $_a$ is vacuous.

Definition 1. *A temporal type* (t_a, i) *is* realisable in S_j *if it satisfies the following conditions.*

1. (t_a, i) *is* DL-realisable in S_j, *that is, if* $\geq n\, r \in t_a$, *then there is a* $(t', i) \in S_j$ *with* $t' \in \mathsf{tp}(\mathcal{K})$ *and* $(\exists r^-) \in t'$*;*
2. (t_a, i) *is* temporally realisable in S_j, *that is, there is a* 2^{cn}*-labelled* $\sharp_\mathbf{E}(\mathcal{K})$*-ary tree* (T, τ) *that satisfies the following*
   (a) *for some* $w \in T$ *at distance* i *from the root* ($|w| = i$)*, we have that* $\tau(w) = \bar{t}_a$*;*
   (b) *for each* $w \in T$*, there is a* $(t'_a, \varrho(|w|)) \in S_j$ *with* $\tau(w) = \bar{t}'_a$*;*

(c) ε *satisfies* $\varphi = \mathbf{A}\Box(\bigwedge_{i=1}^{3}\varphi_i) \wedge \varphi_4 \wedge \varphi_5$ *where*

$$\varphi_1 = \bigwedge_{X_C \in \mathsf{cn}} X_C \leftrightarrow \bar{C} \wedge \bigwedge_{X_\beta \in \mathsf{cn}} X_\beta \leftrightarrow \bar{\beta}$$

$$\varphi_2 = \bigwedge_{X_{r(a,b)} \in \mathsf{cn}} X_{r(a,b)} \wedge (X_{\neg r(a,b)} \vee X_{\neg r^-(b,a)}) \rightarrow \bot$$

$$\varphi_3 = \bigwedge_{r \in \mathsf{Rol}_{\mathsf{rig}}(\mathcal{K}),\, n \in \sharp_{\mathcal{T}}} \mathbf{E}\Diamond X_r^n \rightarrow \mathbf{A}\Box X_r^n$$

$$\varphi_4 = \bigwedge_{r \in \mathsf{Rol}(\mathcal{K}), i \in \sharp_{\mathcal{A}}} \mathbf{A}\bigcirc^i X_r^{n_a^{r,i}}$$

$$\varphi_5 = \bigwedge_{r \in \mathsf{Rol}_{\mathsf{loc}}(\mathcal{K}), X_{\mathbf{E}\bigcirc^i r(a,b)} \in \mathsf{cn}} \mathbf{E}\bigcirc^i (X_{r(a,b)} \wedge X_r^{\hat{n}_a^{r,i}})$$

where $\varrho(i) = \mathtt{min}(i, n_0)$*,* $\sharp_{\mathbf{E}}(\mathcal{K})$ *denotes the number of concepts and assertions of the form* $\mathbf{E}\mathcal{C}$ *and* $\mathbf{E}\bigcirc^i\alpha$ *in* $\mathcal{K}$*, and* $n_a^{r,i}$ *and* $\hat{n}_a^{r,i}$ *are defined as follows: Let* $\mathcal{A}_i^r$ *be the set defined as:*

- $\{r(a,b) \mid \mathbf{P}\bigcirc^j r(a,b) \in \mathcal{A}\}$*, if* $r \in \mathsf{Rol}_{\mathsf{rig}}(\mathcal{K})$*,*
- $\{r(a,b) \mid \mathbf{A}\bigcirc^i r(a,b) \in \mathcal{A}\}\}$*, if* $r \in \mathsf{Rol}_{\mathsf{loc}}(\mathcal{K})$*.*

Then, $n_a^{r,i}$ *is* $\mathtt{max}(\{0\} \cup \{m \in \sharp_{\mathcal{T}}^r \mid r(a,b_1)\ldots r(a,b_m) \in \mathcal{A}_i^r$*, for distinct* $b_1,\ldots,b_m\})$ *and* $\hat{n}_a^{r,i}$ *is* $n_a^{r,i}+1$ *if* $n_a^{r,i}+1 \in \sharp_{\mathcal{T}}^r$*, and* $n_a^{r,i}$*, otherwise.*

Intuitively, Condition 1 takes care of the 'DL-dimension' in the sense that it ensures that each time point will have associated an appropriate DL-interpretation. Condition 2 generally takes care of the temporal evolution of a single domain element a, captured by the tree (T,τ), and of verifying Boolean consistency of types; indeed, that is why types are not required to 'respect' Booleans. More interestingly, Condition $2(c)$ takes care of the temporal dimension by checking that (i) the fresh concept names faithfully represent the surrogated concepts and role assertions they are substituting (φ_1), (ii) no inconsistencies are introduced by role assertions (φ_2), (iii) the semantics of rigid roles is respected, that is, if a fresh concept name standing for a number restriction involving rigid roles is satisfied at some world, then it must be satisfied at all worlds (φ_3) and (iv) r-successors induced by the ABox are properly witnessed (φ_4 and φ_5). The numbers $n_a^{r,i}$ and $\hat{n}_a^{r,i}$, used in (iv), intuitively, take into account the number of r-successors of a induced by $\mathcal{A}$ at worlds at distance i from the root.

The algorithm terminates when $S_j = S_{j+1}$ (no further temporal types are eliminated) and returns "satisfiable" if for every $a \in \mathsf{ind}(\mathcal{A})$, there is a type $(t_a, 0) \in S_j$ such that (i) if $\mathbf{P}\bigcirc^i C(a) \in \mathcal{A}$ (C of the form A or $\neg A$), then $\mathbf{P}\bigcirc^i C \in t_a$ and (ii) if a occurs in a role assertion $\alpha \in \mathcal{A}$, then $\alpha \in t_a$; and "unsatisfiable" otherwise.

Bound n_0**.** It now remains to determine the value of n_0 for which the algorithm is correct. Intuitively, we are looking for an n_0 ensuring that all necessary information to build an (infinite) model is captured in the final result S of the algorithm. In other words, it must be ensured that all the temporal types of the infinite expansion

$$S_\omega = S \cup \{(t,m) \mid (t,n_0) \in S \wedge m > n_0)\}$$

are realisable in the sense that they satisfy Conditions 1 and 2 of Definition 1 when $\varrho(i)$ is replaced with i.

We start by observing that while for CTL-*Lite*$_{bool}^{\mathcal{N}}$ types capture the required information about models, making the number of types an appropriate n_0 bound, they fail to do so for CTL*-*Lite*$_{bool}^{\mathcal{N}}$. Intuitively, this is because CTL*-*Lite*$_{bool}^{\mathcal{N}}$ allows to nest an arbitrary number of temporal operators, as for example in $\mathbf{E}\bigcirc\bigcirc\bigcirc C$, but this is not reflected in the definition of (state) subconcepts and therefore in that of types. To solve this problem, we follow an approach suggested by Gutiérrez-Basulto *et al.* [2012] in which the aforementioned connection between NBTAs and CTL and CTL* is used to show that the states Q of the NBTA $\mathfrak{A}_\varphi$, accepting precisely the 2^{cn}-labelled $\sharp_{\mathbf{E}}(\mathcal{K})$-ary trees satisfying φ in Condition 2(c), ensure that all elements of the infinite expansion of S are realisable; indeed, states do 'memorise' consecutive temporal operators. We thus obtain the following bound $n_0 := |Q|\cdot|\mathsf{tp}(\mathcal{K})|\cdot|\mathsf{atp}(\mathcal{K})|$, where $|Q| \in 2^{2^{\mathsf{poly}(|\varphi|)}}$ if φ is a CTL* formula, and $|Q| \in 2^{\mathsf{poly}(|\varphi|)}$ if φ is a CTL formula [Kupferman and Vardi, 2005; Vardi, 2006]. See the appendix for more details on the bound. Having n_0 at hand:

Theorem 1. *The algorithm returns "satisfiable" iff there is a model of* $\mathcal{K}$*.*

For proving "$\Rightarrow$", we inductively construct a two-dimensional model of $\mathcal{K}$ using the infinite trees (T,τ) witnessing that all temporal types in S_ω satisfy the modified Condition 2. We roughly proceed as follows. We start with fixing the temporal evolutions of all ABox individuals by choosing, for every $a \in \mathsf{ind}(\mathcal{A})$, some $(t_a, 0) \in S_\omega$, and the corresponding infinite tree witnessing Condition 2. For the inductive step, assume that an element d has in its type t at time point w with $|w| = i$ a number restriction 'demand' $\geq n\ r$. In such a case, we add n elements $d_1,\ldots,d_n$ to the domain (if there already some r-successors, we add less). Then, we choose some type (t', i) from S_ω witnessing Condition 1 of Definition 1 for (t,i), and complete the temporal evolution of each d_j according to the infinite tree (T,τ) witnessing (the modified) Condition 2 for (t', i).

Overall Complexity. We finally argue that the algorithm runs in double and single exponential time for CTL*-*Lite*$_{bool}^{\mathcal{N}}$ and CTL-*Lite*$_{bool}^{\mathcal{N}}$, respectively. Clearly, the bound n_0 is in $2^{2^{\mathsf{poly}(|\mathcal{K}|)}}$ for CTL*-*Lite*$_{bool}^{\mathcal{N}}$ and in $2^{\mathsf{poly}(|\mathcal{K}|)}$ for CTL-*Lite*$_{bool}^{\mathcal{N}}$, and the number of steps of the algorithm is in $n_0 \cdot 2^{\mathsf{poly}(|\mathcal{K}|)}$. The number of states of $\mathfrak{A}_\varphi$, used to check Condition $2(c)$, is n_0, and NBTAs to check Conditions $2(a)$ and $2(b)$ can be constructed using at most n_0 states. Thus, the desired result is obtained from the above arguments and the two following facts: (i) a constant number of NBTAs can be intersected with only a polynomial blowup, and (ii) non-emptiness of NBTAs can be checked in quadratic time in the number of states [Vardi, 1998].

Theorem 2. *KB satisfiability is* EXPTIME*-complete for CTL-Lite*$_{bool}^{\mathcal{N}}$ *and* 2EXPTIME*-complete for CTL*-Lite*$_{bool}^{\mathcal{N}}$*.*

Note that CTL-*Lite*$_{bool}^{\mathcal{N}}$ is rather robust in the sense that by just lightening its components better complexity is not immediately obtained. Indeed, the following can be shown using ideas of the EXPTIME-hardness proof for subsumption in the DL $\mathcal{ELI}$ [Baader *et al.*, 2008].

Theorem 3. *KB satisfiability is* EXPTIME*-hard for CTL-Lite$_{horn}$ with only the temporal operators* $\mathbf{E}\bigcirc$ *and* $\mathbf{A}\bigcirc$*.*

4 CTL- & CTL*-*Lite*$^{\mathcal{N}}_{bool}$ Formulas

In this section, we study extensions of CTL-*Lite*$^{\mathcal{N}}_{bool}$ and CTL*-*Lite*$^{\mathcal{N}}_{bool}$ in which temporal operators are not only applied to concepts, but also to Boolean combinations of concept inclusions and ABox assertions. *CTL*-Lite*$^{\mathcal{N}}_{bool}$*-state formulas* φ, ϑ and *CTL*-Lite*$^{\mathcal{N}}_{bool}$*-path formulas* ψ, χ are defined by the following grammar:

$$\begin{array}{lcl} \varphi, \vartheta & ::= & C \sqsubseteq D \mid C(a) \mid r(a,b) \mid \neg\varphi \mid \varphi \wedge \vartheta \mid \mathbf{E}\varphi \\ \psi, \chi & ::= & \varphi \mid \neg\psi \mid \psi \wedge \chi \mid \bigcirc\psi \mid \psi\,\mathcal{U}\,\chi \end{array}$$

where C, D are CTL*-*Lite*$^{\mathcal{N}}_{bool}$-concepts, $\{a, b\} \subseteq \mathsf{N_I}$ and $r \in \mathsf{N_R}$. Given an interpretation $\mathfrak{I}$, a world w in $\mathfrak{I}$ and a path π, the truth relations $\mathfrak{I}, w \models \varphi$ and $\mathfrak{I}, \pi \models \psi$ are defined as expected (cf. appendix); for instance, $\mathfrak{I}, w \models C \sqsubseteq D$ iff $C^{\mathfrak{I},w} \subseteq D^{\mathfrak{I},w}$. We say *CTL*-Lite*$^{\mathcal{N}}_{bool}$*-formula* to refer to a CTL*-*Lite*$^{\mathcal{N}}_{bool}$-state formula (CTL-*Lite*$^{\mathcal{N}}_{bool}$-formulas are defined analogously), and say that $\mathfrak{I}$ is a *model* of a CTL*-*Lite*$^{\mathcal{N}}_{bool}$-formula φ, written $\mathfrak{I} \models \varphi$, iff $\mathfrak{I}, \varepsilon \models \varphi$.

4.1 Satisfiability of Temporal Formulas

We next devise an algorithm for the satisfiability problem of CTL*-*Lite*$^{\mathcal{N}}_{bool}$ and CTL-*Lite*$^{\mathcal{N}}_{bool}$ formulas combining again type-based abstractions with automata-based approaches to temporal reasoning. We obtain 3EXPTIME and 2EXPTIME upper bounds for CTL*-*Lite*$^{\mathcal{N}}_{bool}$- and CTL-*Lite*$^{\mathcal{N}}_{bool}$ formulas, respectively. In contrast to the TDLs in Section 3, for temporal formulas the independence of elements at each $\mathcal{I}_w$ is lost, and therefore one cannot 'separately' reason about the temporal evolution of each domain element. Indeed, the introduction of temporal TBox formulas, e.g. $\mathbf{A}\Diamond\Box(A \sqsubseteq \mathbf{E}\Diamond\exists r)$, forces to see each $\mathcal{I}_w$ as one inseparable entity. As a consequence, in our decision procedure, tree automata run over trees labelled with sets of types (rather than a single type), representing interpretations.

Let φ be a CTL*-*Lite*$^{\mathcal{N}}_{bool}$ or a CTL-*Lite*$^{\mathcal{N}}_{bool}$-formula whose satisfiability is to be decided, and $\mathsf{ind}(\varphi)$, $\mathsf{Rol}(\varphi)$, $\mathsf{CN}(\varphi)$, $\mathsf{Rol_{rig/loc}}(\varphi)$, $\sharp^r_\varphi$ denote the expected sets, see Section 2. We use $\mathsf{cl}(\varphi)$ and $\mathsf{sub}(\varphi)$ to respectively denote the set of all (i) state concepts occurring in φ together with $\{\exists r \mid r \in \mathsf{Rol}(\varphi)\}$, closed under subconcepts and single negation, and (ii) state subformulas of φ, closed under single negation. We assume wlog. that if $r(a,b) \in \mathsf{sub}(\varphi)$, then $r^-(b,a) \in \mathsf{sub}(\varphi)$. A *concept type for* φ is a set $t \subseteq \mathsf{cl}(\varphi)$ such that $\geq\ n\, r \in t$ implies $\geq\ m\, r \in t$, for all $\geq\ n\, r \in \mathsf{cl}(\varphi)$ and $m \in \sharp^r_\varphi$ with $m < n$; and $\mathsf{tp}(\varphi)$ denotes the set of all concept types for φ. A *pointed type for* φ is a pair (t, a) with $t \in \mathsf{tp}(\varphi)$ and $a \in \mathsf{ind}(\varphi)$, and $\mathsf{ptp}(\varphi)$ denotes the set of all pointed types for φ - t_a is used as in the previous section.

Definition 2. *A* quasistate for φ *is a tuple* (S_1, S_2, S_3) *with* $S_1 \subseteq \mathsf{tp}(\varphi)$*,* $S_2 \subseteq \mathsf{ptp}(\varphi)$ *and* $S_3 \subseteq \mathsf{sub}(\varphi)$ *is a* formula type for φ *such that*

1. *For each* $t \in S_1$*, if* $\geq\ n\, r \in t$*, then there is a* $t' \in S_1$ *with* $\exists r^- \in t'$*;*
2. *For each* $a \in \mathsf{ind}(\varphi)$*,* S_2 *contains exactly one pointed type* t_a*;*
3. *If* $(t, a) \in S_2$*, then* $t \in S_1$*;*
4. *For all* $C(a) \in \mathsf{sub}(\varphi)$ *and* $t_a \in S_2$*,* $C(a) \in S_3$ *iff* $C \in t_a$*;*
5. *For all* $a \in \mathsf{ind}(\varphi)$*,* $r \in \mathsf{Rol}(\varphi)$*,* $\geq\ n^r_a\, r \in t_a$*;*
6. *For all* $t \in S_1$*,* $C \sqsubseteq D \in S_3$ *iff* $C \in t$ *implies* $D \in t$*, for all* $C \sqsubseteq D \in \mathsf{sub}(\varphi)$*,*

where $n^r_a = \mathit{max}(\{0\} \cup \{m \in \sharp^r_\varphi \mid r(a,b_1), \ldots, r(a,b_m) \in S_3$ *for distinct* $b_1, \ldots b_m\})$*. A* quasimodel $\mathfrak{Q}$ for φ *is a* $\mathsf{qs}(\varphi)$*-labelled tree (of any outdegree) with* $\mathsf{qs}(\varphi)$ *denoting the set of all quasistates for* φ*.*

Similar to Section 3, we again surrogate the 'DL component' to use the automata machinery for temporal reasoning. For $t \in \mathsf{tp}(\varphi)$, $\bar{t}$ is the result of replacing each $C \in t \setminus \mathsf{CN}(\varphi)$ with a fresh concept name X_C, and ccn denotes the set of all resulting names, including $\mathsf{CN}(\varphi)$. For all $C \in \mathsf{cl}(\varphi)$, $\bar{C}$ denotes the result of replacing in C every subconcept $\geq\ n\, r$ with X^n_r; and $\sqcap$ with $\wedge$. We also surrogate the 'DL-component' at the formula level: For every $\vartheta \in \mathsf{sub}(\varphi)$, $\bar{\vartheta}$ denotes the result of replacing every subformula α of ϑ of the form $C \sqsubseteq D$, $C(a), r(a,b)$ or $\neg r(a,b)$ with a fresh concept name Y_α, and fcn denotes the set of all concept names introduced in this way. We will 'independently' reason about the temporal evolution of concepts and formulas. For $S \subseteq \mathsf{sub}(\varphi)$, $\bar{S}$ refers to $\{\bar{\vartheta} \mid \vartheta \in S\}$. For a quasimodel $\mathfrak{Q}$, we then use $\mathfrak{Q}_3$ to denote the 2^{fcn}-labelled tree obtained by associating each $w \in \mathfrak{Q}$ with the label $\bar{S}_3(w)$.

We next give conditions on quasimodels for φ ensuring they appropriately describe models of φ.

Definition 3. *A quasimodel* $\mathfrak{Q} = (T, \tau)$ *for* φ *is* proper *if:*

1. $\mathfrak{Q}_3 \models \bar{\varphi} \wedge \mathbf{A}\Box(\varphi_1 \wedge \varphi_2)$*, where*

$$\varphi_1 = \bigwedge_{r(a,b) \in \mathsf{sub}(\varphi)} X_{r(a,b)} \wedge (X_{\neg r(a,b)} \vee X_{\neg r^-(b,a)}) \rightarrow \bot$$

$$\varphi_2 = \bigwedge_{r \in \mathsf{Rol_{rig}}(\varphi), r(a,b) \in \mathsf{sub}(\varphi)} \mathbf{E}\Diamond X_{r(a,b)} \rightarrow \mathbf{A}\Box X_{r(a,b)}$$

2. *For all* $w \in T$*,* $\tau(w) = (S_1, S_2, S_3)$ *and all* $t \in S_1$*, there is a* 2^{ccn}*-labelled tree* (T, τ') *such that*
   - *(a)* $\tau'(w) = \bar{t}$*;*
   - *(b) For all* $w' \in T$ *with* $\tau(w') = (S'_1, S'_2, S'_3)$*, there is a* $t' \in S'_1$ *such that* $\tau'(w') = \bar{t'}$*;*
   - *(c)* ε *satisfies* $\vartheta = \mathbf{A}\Box(\vartheta_1 \wedge \vartheta_2)$ *where*

$$\vartheta_1 = \bigwedge_{X_C \in \mathsf{ccn}} X_C \leftrightarrow \bar{C}$$

$$\vartheta_2 = \bigwedge_{r \in \mathsf{Rol_{rig}}(\varphi), n \in \sharp^r_\varphi} \mathbf{E}\Diamond X^n_r \rightarrow \mathbf{A}\Box X^n_r$$

3. *For all* $a \in \mathsf{ind}(\varphi)$*,* $\tau(\varepsilon) = (S_1, S_2, S_3)$ *with* $(t, a) \in S_2$*, there is a* 2^{cnn}*-labelled tree* (T, τ') *with Condition* (c) *as in Point 2 above and*
   - *(a)* $\tau'(\varepsilon) = \bar{t}$*;*

(b) *For all* $w' \in T$ *with* $\tau(w') = (S'_1, S'_2, S'_3)$ *and* $(t', a) \in S'_2$, *we have that* $\tau'(w') = \bar{t}'$.

Intuitively, Condition 1 takes care of $\mathfrak{Q}$ satisfying the temporal formula φ; in particular, it ensures consistency of role assertions and that if a role assertion involving a rigid role is satisfied at some point, then it is satisfied at all time points. Conditions 2 and 3 ensure that each (pointed) type has an appropriate temporal evolution through the quasimodel; the meaning of Conditions $2(c)$ and $3(c)$ is similar to that of the analogous Conditions in Section 3, cf. Definition 1.

We first show that satisfiability of temporal formulas is characterised by the existence of a proper quasimodel:

Lemma 1. *φ is satisfiable iff there is a proper quasimodel for φ.*

As the next step, we will construct a tree automaton $\mathfrak{A}$ that accepts precisely the proper quasimodels for φ. To achieve this, most importantly, $\mathfrak{A}$ will simulate the runs of the NBTAs $\mathfrak{A}_\vartheta$ associated with Condition $2(c)$ in Definition 3. The use of $\mathfrak{A}$ is sanctioned by the fact that the outdegree of proper quasimodels can be bounded, see [Gutiérrez-Basulto, 2013, Lemma 3.22]. Let $\mathfrak{A}_\vartheta = (Q_1, \Sigma, Q_1^0, \delta_1, F_1)$ be the NBTA accepting precisely the 2^{ccn}-labelled $\sharp_{\mathbf{E}}$-ary trees satisfying ϑ, where $\sharp_{\mathbf{E}}$ is the number of state concepts of the form $\mathbf{E}\mathcal{C}$ in φ. We have the following:

Lemma 2. *There is a proper quasimodel for φ iff there is a proper quasimodel of arity $k = |\mathsf{qs}(\varphi)| \cdot |\mathsf{tp}(\varphi)| \cdot |\mathsf{ind}(\varphi)| \cdot Q_1$.*

The tree automaton $\mathfrak{A}$ we construct in the appendix is a two-way alternating tree automaton (2ATA) [Vardi, 1998]. Intuitively, a 2ATA is needed because of two reasons: (i) $\mathfrak{A}$ must be two-way, and therefore allow a predecessor state, because while simulating $\mathfrak{A}_\vartheta$ in $\mathfrak{A}$ the simulation needs to start at an arbitrary world w, however, the original run of $\mathfrak{A}_\vartheta$ starts at the root world ε. Note that this would not be necessary if we assume *expanding domains*, instead of a constant one. (ii) $\mathfrak{A}$ must be alternating, and therefore being able to send several state successors along a branch, because $\mathfrak{A}$ needs to simulate a run of $\mathfrak{A}_\vartheta$ for each type in a given world. Finally, note that the simulated $\mathfrak{A}_\vartheta$ needs to be a NBTA (rather than an ATA) because we must ensure that one state is assigned to each successor - cf. Conditions $2(b)$ and $3(b)$ in Definition 3. Since the definition of $\mathfrak{A}$ is essentially the same as the one previously introduced by Gutiérrez-Basulto *et al.* [2012] for temporal formulas based on $\mathcal{ALC}$ and CTL and CTL*, we differ it to the appendix.

Overall Complexity. (i) Non-emptiness of 2ATAs can be checked in ExpTime in the number of states [Vardi, 1998]. (ii) The number of states of $\mathfrak{A}_\vartheta$, the most dominating in the definition of the states in $\mathfrak{A}$, is in $2^{2^{\mathsf{poly}(|\varphi|)}}$ if ϑ is a CTL* formula and in $2^{\mathsf{poly}(|\varphi|)}$ if ϑ is a CTL formula. We thus obtain:

Theorem 4. *Satisfiability is in 3*ExpTime *for CTL*-Lite$^{\mathcal{N}}_{bool}$- and in 2*ExpTime *for CTL-Lite$^{\mathcal{N}}_{bool}$-formulas.*

4.2 CTL-*Lite*$_{horn}$ Formulas

Finally, we look at CTL-*Lite*$_{horn}$-formulas, the sub-fragment of CTL-*Lite*$^{\mathcal{N}}_{bool}$ in which C, D are CTL-*Lite*$_{horn}$-concepts, and show that this fragment is already 2ExpTime-hard. In particular, we show the following:

Theorem 5. *Satisfiability of CTL-Lite$_{horn}$-formulas with only local roles and,* $\mathbf{A}\bigcirc$ *and* $\mathbf{E}\bigcirc$ *is 2*ExpTime*-hard.*

The proof of Theorem 5 is by reduction of the word problem of an exponentially space bounded alternating Turing machine. We next outline the main ideas of the reduction. (i) The computation tree of an ATM is represented by the temporal evolution of a single individual name a, such that each time point corresponds to a tape cell and a configuration is then represented by exponentially consecutive time points. (ii) To synchronise i-cells in consecutive configurations, the 'content of a' at position i is stored in a fresh *local* r_a-successor and then recovered back to a using temporal TBoxes of the form $\top \sqsubseteq A \vee \top \sqsubseteq \bar{A}$, with A and $\bar{A}$ disjoint, ensuring that *all* domain elements in a given time point share the truth value of A. (iii) Information is transported between neighbouring configurations using binary counters.

From Theorems 4 and 5, the following is obtained:

Theorem 6. *Satisfiability of CTL-Lite$_{horn}$- and CTL-Lite$^{\mathcal{N}}_{bool}$ formulas is 2*ExpTime*-complete.*

5 Conclusions and Future Work

This paper has advanced the understanding of TDLs based on *DL-Lite*$^{\mathcal{N}}_{bool}$ and, CTL* and CTL. In particular, we developed uniform algorithms for both CTL* and CTL-based extensions, providing (mostly tight) elementary upper bounds.

As immediate future work, we will look for tractable fragments. To this aim, we plan to follow in the footsteps of Artale *et al.* [2013; 2014] and analyse clausal fragments of CTL and CTL*. For example, tractable fragments allowing $\mathbf{E}\Diamond$ (arbitrarily used in both sides of concept inclusions) might be identified since, in contrast to $\Diamond$, it does not lead to non-convexity [Gutiérrez-Basulto *et al.*, 2014].

We will also study branching temporal extensions of *DL-Lite* in the OBDA scenario. We are particularly interested in studying the query answering problem over branching temporal *DL-Lite* ontologies. Indeed, it has been argued that in some settings allowing for different versions of data over time might be desirable [Rondogiannis *et al.*, 1998; Chomicki and Toman, 2005].

Beyond TDLs, we will investigate whether our approaches can be extended so as to derive elementary upper bounds for decidable fragments of first-order branching temporal logic.

Acknowledgments

We thank anonymous reviewers for many useful comments and suggestions for improving the paper.

References

[Artale *et al.*, 2007a] Alessandro Artale, Roman Kontchakov, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. Temporalising tractable description logics. In *Proc. of TIME-07*, 2007.

[Artale *et al.*, 2007b] Alessandro Artale, Carsten Lutz, and David Toman. A description logic of change. In *Proc. of IJCAI-07*, 2007.

[Artale *et al.*, 2009] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The DL-Lite family and relations. *J. Artif. Intell. Res. (JAIR)*, 36:1–69, 2009.

[Artale *et al.*, 2011] Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev. Tailoring temporal description logics for reasoning over temporal conceptual models. In *Proc. of FroCos-11*, 2011.

[Artale *et al.*, 2013] Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev. The complexity of clausal fragments of LTL. In *Proc. of LPAR-19*, 2013.

[Artale *et al.*, 2014] Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev. A cookbook for temporal conceptual data modelling with description logics. *ACM Trans. Comput. Log.*, 15(3):25, 2014.

[Baader *et al.*, 2008] Franz Baader, Carsten Lutz, and Sebastian Brandt. Pushing the EL envelope further. In *Proc. of OWLED-08.*, 2008.

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

[Chomicki and Toman, 2005] Jan Chomicki and David Toman. Temporal Databases. In *Handbook of Temporal Reasoning in Artificial Intelligence*, pages 429–468. Elsevier, 2005.

[Clarke and Emerson, 1981] Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In Dexter Kozen, editor, *Logic of Programs*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer, 1981.

[Crowe and Tao, 2015] Christopher L. Crowe and Cui Tao. Designing ontology-based patterns for the representation of the time-relevant eligibility criteria of clinical protocols. *AMIA Summits on Translational Science Proceedings*, pages 173–177, 2015.

[de Matos Galante *et al.*, 2005] Renata de Matos Galante, Clesio Saraiva dos Santos, Nina Edelweiss, and Álvaro Freitas Moreira. Temporal and versioning model for schema evolution in object-oriented databases. *Data Knowl. Eng.*, 53(2):99–128, 2005.

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

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

[Golendziner and dos Santos, 1995] Lia Goldstein Golendziner and Clesio Saraiva dos Santos. Versions and configurations in object-oriented database systems: A uniform treatment. In *COMAD*, 1995.

[Gutiérrez-Basulto *et al.*, 2012] Víctor Gutiérrez-Basulto, Jean C. Jung, and Carsten Lutz. Complexity of branching temporal description logics. In *Proc. of ECAI-12*, 2012.

[Gutiérrez-Basulto *et al.*, 2014] Víctor Gutiérrez-Basulto, Jean C. Jung, and Thomas Schneider. Lightweight description logics and branching time: a troublesome marriage. In *Proc. of KR-14*, 2014.

[Gutiérrez-Basulto, 2013] Víctor Gutiérrez-Basulto. *Towards More Useful Description Logics of Time, Change and Context*. PhD thesis, University of Bremen, 2013.

[Hodkinson *et al.*, 2002] Ian M. Hodkinson, Frank Wolter, and Michael Zakharyaschev. Decidable and undecidable fragments of first-order branching temporal logics. In *Proc. of LICS-02*, 2002.

[Kupferman and Vardi, 2005] Orna Kupferman and Moshe Y. Vardi. Safraless decision procedures. In *Proc. of FOCS-05*, 2005.

[Kupferman *et al.*, 2000] Orna Kupferman, Moshe Y. Vardi, and Pierre Wolper. An automata-theoretic approach to branching-time model checking. *J. ACM*, 47(2):312–360, 2000.

[Lutz *et al.*, 2008] Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. Temporal description logics: A survey. In *Proc. of TIME-08*, 2008.

[Moro *et al.*, 2001] Mirella Moura Moro, Silvia Maria Saggiorato, Nina Edelweiss, and Clesio Saraiva dos Santos. Dynamic systems specification using versions and time. In *Proc. of IDEAS-01*, pages 99–107, 2001.

[Noronha *et al.*, 1998] Marilene Araujo Noronha, Lia Goldstein Golendziner, and Clesio Saraiva dos Santos. Extending A structured document model with version control. In *Proc. of IDEAS-98*, pages 234–242, 1998.

[O'Connor *et al.*, 2009] Martin J. O'Connor, Ravi D. Shankar, David B. Parrish, and Amar K. Das. Knowledge-data integration for temporal reasoning in a clinical trial system. *I. J. Medical Informatics*, 78:S77–S85, 2009.

[Pratt, 1979] Vaughan Pratt. Models of program logics. In *Proc. of 20th Annual Symposium on Foundations of Computer Science*, pages 115–122, 1979.

[Rondogiannis *et al.*, 1998] Panos Rondogiannis, Manolis Gergatsoulis, and Themis Panayiotopoulos. Branching-time logic programming: The language *Cactus* and its applications. *Comput. Lang.*, 24(3):155–178, 1998.

[Shankar *et al.*, 2008] Ravi D. Shankar, Susana B. Martins, Martin J. O'Connor, David B. Parrish, and Amar K. Das. Representing and reasoning with temporal constraints in clinical trials using semantic technologies. In *BIOSTEC-08, Revised Selected Papers*, pages 520–530, 2008.

[Vardi and Stockmeyer, 1985] Moshe Y. Vardi and Larry J. Stockmeyer. Improved upper and lower bounds for modal logics of programs: Preliminary report. In Robert Sedgewick, editor, *STOC*, pages 240–251. ACM, 1985.

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

[Vardi, 1998] Moshe Y. Vardi. Reasoning about the past with two-way automata. In *In Proc. of ICALP-98*, pages 628–641, 1998.

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

[Weitl *et al.*, 2009] Franz Weitl, Mirjana Jaksic, and Burkhard Freitag. Towards the automated verification of semi-structured documents. *Data Knowl. Eng.*, 68(3):292–317, 2009.

APPENDIX

CTL* and CTL

Syntax. Fix a countably infinite set of *propositional variables* AP. CTL* *state formulas* φ and CTL* *path formulas* ψ are defined by the following grammar:

$$\varphi \quad ::= \quad p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathbf{E}\psi$$
$$\psi \quad ::= \quad \varphi \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \bigcirc\psi \mid \Box\psi \mid \psi_1 \mathcal{U} \psi_2$$

where p ranges over AP, $\varphi, \varphi_1, \varphi_2$ are state formulas, and ψ, ψ_1, ψ_2 are path formulas. Without further quantification, a CTL* *formula* is a state formula. In CTL* we can define Boolean abbreviations standardly, plus the following temporal abbreviations: $\mathbf{A}\varphi = \neg\mathbf{E}\neg\varphi$, $\Diamond\psi = \neg\Box\neg\psi$.

CTL is the *fragment* of CTL* in which temporal operators $\bigcirc, \Box$ and $\mathcal{U}$ must be *immediately* preceded by the path quantifier $\mathbf{E}$. Formally, CTL *state formulas* φ and CTL *path formulas* ψ are defined by the following grammar:

$$\varphi \quad ::= \quad p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathbf{E}\psi$$
$$\psi \quad ::= \quad \bigcirc\varphi \mid \Box\varphi \mid \varphi_1 \mathcal{U} \varphi_2$$

where p ranges over AP, $\varphi, \varphi_1, \varphi_2$ are state formulas, and ψ is a path formula. In CTL temporal abbreviations are defined as follows:

$$\mathbf{A}\bigcirc\varphi = \neg\mathbf{E}\bigcirc\neg\varphi, \qquad \mathbf{E}\Diamond\varphi = \mathbf{E}(\mathsf{true}\,\mathcal{U}\varphi),$$
$$\mathbf{A}\Box\varphi = \neg\mathbf{E}\Diamond\neg\varphi, \qquad \mathbf{A}\Diamond\varphi = \neg\mathbf{E}\Box\neg\varphi,$$
$$\mathbf{A}(\varphi_1 \,\mathcal{U}\, \varphi_2) = \neg\mathbf{E}(\neg\varphi_2 \,\mathcal{U}\, (\neg\varphi_1 \wedge \neg\varphi_2)) \wedge \neg\mathbf{E}\Box\neg\varphi_2.$$

Trees. 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 $E \subseteq W \times W$ is a total relation. We say that wc is a *successor* of w, and that 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. Furthermore, we say that T is a k*-ary tree*, $k \geq 1$ if every node of T has exactly k successors. Let Σ be a finite alphabet. A Σ*-labeled tree* $\mathfrak{T}$ is a pair (T, τ) where T is a tree and $\tau : T \to \Sigma$ assigns a letter from Σ to each node. We sometimes identify (T, τ) with τ.

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 exactly one $c \in \mathbb{N}$ with $w'c \in \pi$. We use $\mathsf{Paths}(w)$ to denote the set of all paths starting at the node w; and for a path $\pi = w_0 w_1 w_1 \cdots$ and $i \geq 0$, we use $\pi[i]$ to denote w_i and $\pi[i..]$ to denote the path $w_i w_{i+1} \cdots$.

Semantics. To define the semantics of CTL*, we consider Σ-labeled trees with $\Sigma = 2^{\mathsf{AP}}$. Intuitively, the label of a time point contains the propositional letters holding at this time point.

Let $\mathfrak{T} = (T, \tau)$ be a 2^{AP}-labeled tree. For a time point w in $\mathfrak{T}$, the *truth relation* $\models$ for CTL* state formulas is defined as follows.

$\mathfrak{T}, w \models p \in \mathsf{AP}$	iff	$p \in \tau(w)$;
$\mathfrak{T}, w \models \neg\varphi$	iff	$\mathfrak{T}, w \not\models \varphi$;
$\mathfrak{T}, w \models \varphi_1 \wedge \varphi_2$	iff	$\mathfrak{T}, w \models \varphi_1$ and $\mathfrak{T}, w \models \varphi_2$;
$\mathfrak{T}, w \models \mathbf{E}\psi$	iff	$\mathfrak{T}, \pi \models \psi$ for some $\pi \in \mathsf{Paths}(w)$.

For a path π in $\mathfrak{T}$, the truth relation $\models$ for path formulas is defined as follows:

$\mathfrak{T}, \pi \models \varphi$	iff	$\mathfrak{T}, \pi[0] \models \varphi$;
$\mathfrak{T}, \pi \models \neg\psi$	iff	$\mathfrak{T}, \pi \not\models \psi$;
$\mathfrak{T}, \pi \models \psi_1 \wedge \psi_2$	iff	$\mathfrak{T}, \pi \models \psi_1$ and $\mathfrak{T}, \pi \models \psi_2$;
$\mathfrak{T}, \pi \models \bigcirc\psi$	iff	$\mathfrak{T}, \pi[1..] \models \psi$;
$\mathfrak{T}, \pi \models \Box\psi$	iff	$\forall j \geq 0. \mathfrak{T}, \pi[j..] \models \psi$;
$\mathfrak{T}, \pi \models \psi_1 \,\mathcal{U}\, \psi_2$	iff	$\exists j \geq 0. (\,\mathfrak{T}, \pi[j..] \models \psi_2 \wedge$ $\forall 0 \leq k < j. (\,\mathfrak{T}, \pi[k..] \models \psi)\,)$.

Reasoning Problems. In CTL*, as in DLs, one of the classical reasoning problems is the *satisfiability problem*: a 2^{AP}-labeled tree $\mathfrak{T}$ is a *model of a* CTL* *formula* φ if $\mathfrak{T}, \varepsilon \models \varphi$. A CTL* formula φ is *satisfiable* if there exists a 2^{AP}-labeled tree $\mathfrak{T}$ such that $\mathfrak{T}$ is a model of φ.

Nondeterministic Tree Automata

A nondeterministic Büchi tree automaton (NBTA) over Σ-labeled k-ary trees is a tuple $\mathfrak{A} = (Q, \Sigma, Q^0, \delta, F)$ where Q is a finite set of *states*, $Q^0 \subseteq Q$ is the set of *initial states*, $F \subseteq Q$ is a set of *recurring states*, and $\delta : Q \times \Sigma \to 2^{Q^k}$ is the *transition function*. A *run* of $\mathfrak{A}$ on τ is a Q-labeled k-ary tree (T, r) such that $r(\varepsilon) \in Q^0$ and for each node $w \in T$, we have $\langle r(w \cdot 1), \ldots, r(w \cdot k) \rangle \in \delta(r(w), \tau(w))$. The run is *accepting* if for every path $\pi = w_0 w_1 \cdots$ which starts at ε, we have $r(w_i) \in F$ for infinitely many i. The set of trees accepted by $\mathfrak{A}$ is denoted by $L(\mathfrak{A})$. The *emptiness-problem* is the following: given a NBTA $\mathfrak{A}$, determine whether $L(\mathfrak{A}) \neq \emptyset$. The emptiness-problem for NBTAs can be decided in quadratic time in the number of states [Vardi and Wolper, 1986].

For $n > 0$, we use $\mathsf{Mod}_n(\varphi)$ to denote the set of all n-ary models of φ, and $\mathsf{ap}(\varphi)$ to denote the set of atomic propositions in a CTL* formula φ. The following property shows that it is sufficient to only consider models of certain arity.

Proposition 1 ([Kupferman *et al.*, 2000]). *A CTL* formula 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 $\mathbf{E}\psi$.*

We now assert the precise relation between the satisfiability problem for temporal logics and the nonemptiness problem for NBTAs.

Theorem 7 ([Kupferman and Vardi, 2005; Vardi, 2006]). *For a CTL*-formula φ and $n \geq 0$, one can construct an NBTA $\mathfrak{A}_\varphi = (Q, \Sigma, \delta, Q^0, F)$ in time $\mathsf{poly}(|Q| + n)$ such that $L(\mathfrak{A}_\varphi) = \mathsf{Mod}_n(\varphi)$, $\Sigma = 2^{\mathsf{ap}(\varphi)}$, $|Q| \in 2^{2^{\mathsf{poly}(|\varphi|)}}$, and $|Q| \in 2^{\mathsf{poly}(|\varphi|)}$ when φ is a CTL formula.*

Note that Theorem 7 admits any outdegree.

Proofs for Section 3

Before showing the correctness of our algorithm, cf. Theorem 1, we describe the strategy to show that the proposed n_0 bounds in Section 3 guarantee the correctness of the algorithm. Since we want to guarantee that the the infinite expansion S_ω of the result S of the algorithm conforms with the realisability conditions, we show the following property.

Theorem 8. *Let Q be the states of the NBTA $\mathfrak{A}_\varphi$ associated with φ in Condition 2(c). If $n_0 := |Q|\cdot|\mathsf{atp}(\mathcal{K})|\cdot|\mathsf{etp}(\mathcal{K})|$, then the following holds:*

- *(t, n_0) is realisable in S implies $(t, n_0 + \ell)$ is realizable in S for all $\ell > 0$*

Proof Sketch. We next describe the strategy of the proof. The details of the intermediate results can be proved using the techniques developed for CTL$_{\mathcal{ALC}}$ and CTL$^*_{\mathcal{ALC}}$ [Gutiérrez-Basulto *et al.*, 2012], see [Gutiérrez-Basulto, 2013, Lemma 3.6-3.9] for details. As discussed, we make use of the states Q of the automaton $\mathfrak{A}_\varphi$, with φ as in Condition $2(c)$ of Definition 1; naturally, for the final result of the algorithm we are interested on those states occuring in accepting run of $\mathfrak{A}_\varphi$. We thus have the following.

Definition 4. *An* extended temporal type for $\mathcal{K}$ *is a triple (t, q, i) with (t, i) a temporal type for $\mathcal{K}$ and $q \in Q$. Let $\widehat{S}$ be the set of all extended temporal types such that $(t, i) \in S$ and there is a 2^{cn}-labeled tree and an accepting run (T, r) of $\mathfrak{A}_\varphi$ such that the following holds*

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

It is not hard to see that $\widehat{S}$ satisfies an 'extended' version of the realisability conditions in Definition 1. Note that $\widehat{S}$ inherently satisfies Condition $2(c)$. We thus have the following (the subindex $_a$ is vacuous if t is a concept type).

Proposition 2. *For all (t_a, q, i) the following holds:*

$\widehat{1}$ *If $(\geq n\ r) \in t_a$, there is a $(t, i) \in \widehat{S}$ such that $(\exists r^-) \in t$;*

$\widehat{2}$ *There is a 2^{cn}-labeled tree (T, τ) and an accepting run (T, r) of $\mathfrak{A}_\varphi$ on (T, τ) such that*

- *(a) for some $w \in T$ at distance i from the root ($|w| = i$), we have that $\tau(w) = \bar{t}_a$ and $r(w) = q$;*
- *(b) for each $w \in T$, there is a $(t'_a, q', \varrho(|w|)) \in S_j$ with $\tau(w) = \bar{t}'_a$ and $r(w) = q'$.*

The key result to show the desired property of Theorem 8 above is that the temporal types in $\widehat{S}$ exhibit a monotonic behaviour in the sense that over time we can only lose extended types or get stable. This result, intuitively, holds due to the 'light interaction' between the component logics, and therefore the high degree of independence of the elements in each $\mathcal{I}_w$. We note in passing that this has also been observed in combinations, with similar design choices, based on LTL [Lutz *et al.*, 2008, Lemma 15].

Lemma 3. *For all $i \leq n_0$, let $\mathfrak{S}_i = \{(t, q) \mid (t, q, i) \in \widehat{S}\}$. We have that*

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

With Lemma 3 at hand, it is not hard to see that the infinite continuation $\widehat{S_\omega}$ of $\widehat{S}$ is realizable. More precisely, $\widehat{S_\omega} = \{(t, q, i) \mid (t, q, \varrho(i)) \in \widehat{S}\}$ satisfies conditions $\widehat{1}'$ and $\widehat{2}'$, which are as in Proposition 2, but $i \in \mathbb{N}$ is allowed and $\varrho(i)$ is replaced with i in $2(b)$.

We are now ready to finish the proof:

We define conditions 1' and 2' as variants of Conditions 1 and 2 in Definition 1 by admitting every $i \in \mathbb{N}$ and replacing $\rho(i)$ with i in Condition 2. Let $(t, n_0) \in S$, then, by definition of $\widehat{S}$, there is some q such that $(t, q, n_0) \in \widehat{S}$. We know $(t, q, n_0 + \ell) \in \widehat{S}_\omega$ 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 $\widehat{S}_\omega$, $(t', q', n_0) \in \widehat{S}_\omega$, thus $(t, n_0) \in S$ and $(t, n_0 + \ell) \in S_\omega$. Hence, condition 1' is satisfied for $(t, n_0 + \ell)$. It can be shown that $(t, n_0 + \ell)$ satisfies Condition 2' in analogous way. Therefore, $(t, n_0 + \ell)$ is realizable in S.

This completes the proof of Theorem 8.

□

We are now ready to show the correctness of the algorithm.

Theorem 1 *The algorithm returns 'satisfiable' iff there is a model of $\mathcal{K}$.*

Proof. "⇒" Let S be the final result of the algorithm; and fix $k = \sharp_{\mathbf{E}}(\mathcal{K})$. Recall that due to Theorem 8 above, for every $(t, i) \in S_\omega$ there is a 2^{cn}-labeled k-ary tree $(T, \tau_{t,i})$ satisfying Condition 2 φ (replacing $\varrho(i)$ with i) in Definition 1. We construct a model $\mathfrak{I} = (\Delta^{\mathfrak{I}}, (\mathcal{I}_w)_{w \in W})$ of $\mathcal{K}$, where the underlying (infinite) tree $T = (W, E)$ is a k-ary tree.

Before proceeding with the construction, we make the following observations, which will be used later on.

Observation 1. *For all $(t, i) \in S_\omega$ and $w \in W$, if $X^r_n \in \tau_{t,i}(w)$ then*

- *there is a $(t', j) \in S_\omega$ such that with $X_{\exists r^-} \in \tau_{t',j}(w)$ and $\tau_{t',j}$ witness Condition 2 in Definition 1 for (t', j) .*

In this case we say that $\tau_{t',j}$ is r-compatible with $\tau_{t,i}$ in w.

The observation follows from these facts: (i) since all $(t, i) \in S_\omega$ are realizable (in particular, they fulfil the DL-Condition), there is a $(t', j) \in S_\omega$ such that $X_{\exists r^-} \in \tau_{t',j}(w')$ and $|w| = |w'|$ and, moreover, $\tau_{t',j}$ witness Condition 2 for (t', j); and (ii) the desired $\tau_{t',j}$ can be obtained from that in (i) by permuting successors. In particular, it is not hard to see that if there is an accepting run of $\mathfrak{A}_\varphi$ over $\tau_{t',j}$ as in (i), then there is accepting run over the 'permuted' one. Indeed, the transitions of $\mathfrak{A}_\varphi$ are closed under permuting successors.

Observation 2. *Following the arguments above, we also assume wlog. that for all ABox types $(t_a, i) \in S_\omega$ and $w \in W$, if $X_{r(a,b)} \in \tau_{t_a,0}(w)$ then $X_{r^-(b,a)} \in \tau_{t_b,0}(w)$.*

To construct $\mathfrak{I}$, we define sequences $\Delta_0, \Delta_1, \ldots$ and partial mappings $\pi_i : \Delta_i \to S$ and $\theta_i : \Delta_i \times W \to 2^{\mathsf{cn}}$, and relations R_0^r, R_1^r with $r \in \mathsf{Rol}(\mathcal{K})$.

We start the construction of $\mathfrak{I}$ as follows. For each $a \in \mathsf{ind}(\mathcal{A})$, let $(t_a, 0)$ be a type fulfilling the 'satisfiable' requirement when the algorithm terminates. Then set:

- $\Delta_0 = \mathsf{ind}(\mathcal{A})$;
- $\pi(a) = (t_a, 0)$, for all $a \in \Delta_0$;
- $\theta(a, w) = \tau_{t_a,0}(w)$, for all $a \in \Delta_0$ and $w \in W$;
- If $X_{r(a,b)} \in \tau_{t_a,0}(w)$ proceed as follows:
  - If $r \in \mathsf{Rol}_{\mathsf{loc}}(\mathcal{K})$, add (a, b, w) to R_0^r;
  - If $r \in \mathsf{Rol}_{\mathsf{rig}}(\mathcal{K})$, add (a, b, w') to R_0^r, for all $w' \in W$.

We now introduce some required definitions to continue with the inductive steps. The *required r-rank for a $d \in \Delta_i, i \geq 1$, at world w and step i* is defined as follows:

$$\sharp_{d,i}^{r,w} = \mathtt{max}\{n \in \sharp_{\mathcal{T}}^r \mid X_n^r \in \theta_i(d, w)\}^2$$

The *actual r-rank for a $d \in \Delta_i, i \geq 1$, at world w and step i* is defined as follows:

$$\xi_{d,i}^{r,w} = \mathtt{max}\{\{0\} \cup \{n \in \sharp_{\mathcal{T}}^r \mid \exists \text{ distinct } e_1, \ldots e_n \in \Delta_i \wedge (d, e_j) \in R_i^r(w) \text{ or } (e_j, d) \in R_i^{r^-}(w)\}\}$$

For the inductive step, set $\Delta_i = \Delta_{i-1}$, $\pi_i = \pi_{i-1}$, $\theta_i = \theta_{i-1}$, $R_i^r = R_{i-1}^r$ and apply the following, which is meant to fix 'defective' points missing successors:

- If $\sharp_{d,i}^{r,w} - \xi_{d,i}^{r,w} = n > 0$ for some $d \in \Delta_{i-1}$ and $w \in W$ and $r \in \mathsf{Rol}(\mathcal{K})$, then proceed as follows:
  1. Add $e_1, \ldots, e_n$ to Δ_i;
  2. Add (d, e_k, w) to R_i^r if r is local; otherwise, for all $w' \in W$, add (d, e_k, w') to R_i^r;
  3. For all $w' \in W$, set $\theta_i(e_k, w') ::= \tau_{(t,j)}(w')$ such that $\tau_{t,j}$ is r-compatible with $\tau_{\pi_i(d)}$ in w; and set $\pi_i(e_k) ::= (t, j)$.

Finally, put $\Delta^{\mathfrak{I}} = \bigcup_{i \geq 0} \Delta_i$, $\pi = \bigcup_{i \geq 0} \pi_i$, $\theta = \bigcup_{i \geq 0} \theta_i$ and $R^r = \bigcup_{i \geq 0} R_i^r$. It remains to define the interpretation of concept and role names (recall we make the standard name assumption):

$$\begin{array}{rcl} A^{\mathfrak{I},w} & = & \{d \in \Delta^{\mathfrak{I}} \mid A \in \theta(d, w)\}; \\ s^{\mathfrak{I},w} & = & \{(d, e) \in (\Delta^{\mathfrak{I}})^2 \mid (d, e, w) \in R^s\} \cup \\ & & \{(d, e) \in (\Delta^{\mathfrak{I}})^2 \mid (e, d, w) \in R^{s^-}\} \end{array}$$

For finishing the "⇒" we use the following claim, which can be proved by simultaneous induction on the structure of C and $\mathcal{C}$.

Claim. For all $d \in \Delta^{\mathfrak{I}}$, $w \in W$ and $C \in \mathsf{ccl}(\mathcal{K})$ we have that

$$C \in \theta(d, w) \text{ iff } d \in C^{\mathfrak{I},w},$$

and for every $\pi \in \mathsf{Paths}(w)$ and path concept $\mathcal{C}$

$$d \in \mathfrak{C}^{\mathfrak{I},\pi} \text{ iff } \tau_{\pi(d)} \models \bar{\mathfrak{C}}$$

Proof of Claim. (i) Let $d \notin \mathsf{ind}(\mathcal{A})$, $\pi(d) = (t, i)$ and $\theta(d, w) = \tau_{t,i}(w)$ for some $(t, i) \in S_\omega$.

- $C = \neg D$ "if:" $d \in \neg D^{\mathfrak{I},w}$, that is, $d \notin D^{\mathfrak{I},w}$. Now, by I.H., $X_D \notin \tau_{t,i}(w)$. Furthermore, by Condition $2'(c)$, $(T, \tau_{t,i}), w \models \neg\overline{D}$. Finally, again by Condition $2'(c)$, $X_{\neg D} \in \tau_{t,i}(w)$.
  "only if:" $X_{\neg D} \in \tau_{t,i}(w)$, by Condition $2'(c)$, $(T, \tau_{t,i}), w \not\models \overline{D}$. Now, by Condition $2'(c)$, $X_D \notin \tau_{t,i}(w)$. By, I.H., $d \notin D^{\mathfrak{I},w}$. Therefore, $d \in (\neg D)^{\mathfrak{I},w}$.
- $C = D \sqcap E$ "if:" $d \in (D \sqcap E)^{\mathfrak{I},w}$, that is, $d \in D^{\mathfrak{I},w}$ and $d \in E^{\mathfrak{I},w}$. By I.H., $X_D \in \tau_{t,i}(w)$ and $X_E \in \tau_{t,i}(w)$. Now, by Condition $2'(c)$, $(T, \tau_{t,i}), w \models \overline{D}$ and $(T, \tau_{t,i}), w \models \overline{E}$. So, $(T, \tau_{t,i}), w \models \overline{D} \wedge \overline{E}$. Once again, by Condition $2'(c)$, $(T, \tau_{t,i}), w \models X_{D \sqcap E}$. Therefore, $X_{D \sqcap E} \in \tau_{t,i}(w)$.
  "only if:" $X_{D \sqcap E} \in \tau_{t,i}(w)$, by Condition $2'(c)$, we have that $(T, \tau_{t,i}), w \models \overline{D} \wedge \overline{E}$, that is, $(T, \tau_{t,i}), w \models \overline{D}$ and $(T, \tau_{t,i}), w \models \overline{E}$. Once again, by Condition $2'(c)$, we have that $X_D, X_E \in \tau_{t,i}(w)$. Now, by I.H., $d \in D^{\mathfrak{I},w}$ and $(t, i) \in d^{\mathfrak{I},w}$. Therefore, $d \in (D \sqcap E)^{\mathfrak{I},w}$.
- $C = \geq\ n\ r$
  - *r is local:* $d \in (\geq\ n\ r)^{\mathfrak{I},w}$ *iff* $\sharp\{e \mid (d, e) \in r^{\mathfrak{I},w}\} \geq n$. *iff* , by definition of $\mathfrak{I}$, $\sharp\{e \mid (d, e) \in R^r(w) \text{ or } (e, d) \in R^{r^-}(w)\} \geq n$ *iff* there is a step j in the construction in which the inductive rule was applied to d for some $X_m^r \in \theta(d, w)$ with maximal $m \geq n$ and $X_m^r \in \theta(d, w)$ *iff* , by second condition of type, $X_n^r \in \theta(d, w)$.
  - *r is rigid:* $d \in (\geq\ n\ r)^{\mathfrak{I},w}$ *iff* $\sharp\{e \mid (d, e) \in r^{\mathfrak{I},w}\} \geq n$. *iff* , by definition of $\mathfrak{I}$, $\sharp\{e \mid (d, e) \in R^r(w) \text{ or } (e, d) \in R^{r^-}(w)\} \geq n$ *iff* there is a step j in the construction in which the inductive rule was applied to d for some $X_m^r \in \theta(d, w')$ with $w' \in W$ and maximal $m \geq n$ and $X_m^r \in \theta(d, w')$ *iff* , by second condition of type, $X_n^r \in \theta(d, w')$ *iff*, by φ_3 in $2(c)$, $X_n^r \in \tau_{t,i}(w) = \theta(d, w)$.
- $C = \mathbf{E}\mathfrak{C}$ "if:" $d \in (\mathbf{E}\mathfrak{C})^{\mathfrak{I},w}$. This implies that, by semantics, $d \in \mathfrak{C}^{\mathfrak{I},\pi}$ for some $\pi \in \mathsf{Paths}(w)$. Now, by the second point of the claim, $(T, \tau_{t,i}), \pi \models \overline{\mathfrak{C}}$. Therefore, by semantics, $(T, \tau_{t,i}), w \models \overline{\mathbf{E}\mathfrak{C}}$. Since, $(T, \tau_{t,i}) \models \varphi$ from Condition $2'(c)$, then $X_{\mathbf{E}\mathfrak{C}} \in \tau_{t,i}(w)$.
  "only if:" $X_{\mathbf{E}\mathfrak{C}} \in \tau_{t,i}(w) = \overline{t'}$. By Condition $2'(c)$ we have that $(T, \tau_{t,i}), w \models \overline{\mathbf{E}\mathfrak{C}}$, that is, $(T, \tau_{t,i}), \pi \models \overline{\mathfrak{C}}$ for some $\pi \in \mathsf{Paths}(w)$. Now, by the second point of the claim, $(t, i) \in \mathfrak{C}^{\mathfrak{I},\pi}$. Therefore, $(t, i) \in (\mathbf{E}\mathfrak{C})^{\mathfrak{I},w}$.

This finishes the proof of the first point of the claim.

We proceed to show the second point of the claim:

- $\mathfrak{C} = D$ with D a state concept. "if:" $d \in D^{\mathfrak{I},\pi[0]}$. Note that $\pi[0] = w$, then, by the first point of the claim, $(T, \tau_{t,i}), w \models \overline{D}$.

[2] By convention $n = 0$ if no such proposition is in $\theta(d, w)$

"only if:" $(T, \tau_{t,i}), \pi[0] \models \overline{D}$. By Condition $2'(c)$ $X_D \in \tau_{t,i}(\pi[0])$. Note that $\pi[0] = w$, then, by first point of the claim, $d \in D^{\mathfrak{I},w}$.

- $\mathfrak{C} = \neg\mathfrak{D}$ and $\mathfrak{C} = \mathfrak{C}_1 \sqcap \mathfrak{C}_2$, similar to the analogous case for state concepts.
- $\mathfrak{C} = \bigcirc\mathfrak{D}$. "if:" $d \in (\bigcirc\mathfrak{D})^{\mathfrak{I},\pi}$, that is, $d \in \mathfrak{D}^{\mathfrak{I},\pi[1]}$. Now, by I.H., $(T, \tau_{t,i}), \pi[1] \models \overline{\mathfrak{D}}$. Therefore, by semantics, $(T, \tau_{t,i}), \pi \models \bigcirc\overline{\mathfrak{D}}$.

  "only if:" $(T, \tau_{t,i}), \pi \models \bigcirc\overline{\mathfrak{D}}$. Hence, $(T, \tau_{t,i}), \pi[1] \models \overline{\mathfrak{D}}$. Now, by I.H., $d \in \mathfrak{D}^{\mathfrak{I},\pi[1]}$. Therefore, by semantics, $d \in (\bigcirc\mathfrak{D})^{\mathcal{I},\pi}$.
- $\mathfrak{C} = \Box\mathfrak{D}$. "if:" $d \in (\Box\mathfrak{D})^{\mathfrak{I},\pi}$, that is, for all $j \geq 0$, $d \in \mathfrak{D}^{\mathfrak{I},\pi[j..]}$. Now, by I.H., $(T, \tau_{t,i}), \pi[j..] \models \overline{\mathfrak{D}}$. Therefore, by semantics, $(T, \tau_{t,i}), \pi \models \Box\overline{\mathfrak{D}}$.

  "only if:" $(T, \tau_{t,i}), \pi \models \Box\overline{\mathfrak{D}}$. This means that for all $j \geq 0$, $(T, \tau_{t,i}), \pi[j..] \models \overline{\mathfrak{D}}$. Now, by I.H., for all $j \geq 0$, $d \in \mathfrak{D}^{\mathfrak{I},\pi[j..]}$. Therefore, by semantics, $d \in (\Box\mathfrak{D})^{\mathfrak{I},\pi}$.
- $\mathfrak{C} = \mathfrak{C}_1 \mathcal{U} \mathfrak{C}_2$. "if" $\exists j \geq 0.(d \in \mathfrak{C}_2^{\mathfrak{I},\pi[j..]} \wedge \forall 0 \leq k < j.d \in \mathfrak{C}_1^{\mathfrak{I},\pi[k..]})$. Now, by I.H., $(T, \tau_{t,i}), \pi[j..] \models \overline{\mathfrak{C}}_2 \wedge \forall 0 \leq k < j.((T, \tau_{t,i}), \pi[k..] \models \overline{\mathfrak{C}}_1)$. Therefore, $(T, \tau_{t,i}), \pi \models \overline{(\mathfrak{C}_1 \mathcal{U} \mathfrak{C}_2)}$.

  "only if:" $\exists j \geq 0.((T, \tau_{t,i}), \pi[j..] \models \overline{\mathfrak{C}}_2 \wedge \forall 0 \leq k < j.((T, \tau_{t,i}), \pi[k..] \models \overline{\mathfrak{C}}_1)$. Now, by I.H., $d \in \mathfrak{C}_2^{\mathfrak{I},\pi[j..]} \wedge \forall 0 \leq k < j.d \in \mathfrak{C}_1^{\mathfrak{I},\pi[k..])}$. Therefore, $d \in (\mathfrak{C}_1 \mathcal{U} \mathfrak{C}_2)^{\mathfrak{I},\pi}$.

  This finishes the proof of the second point of the claim.

(ii) Let $a \in \mathsf{ind}(\mathcal{A})$, $\pi(a) = (t_a, 0)$ and $\theta(a, w) = \tau_{t_a,0}(w)$ for some $(t, i) \in S_\omega$. All the cases work are exactly as above, except for $\geq\ n\ r$, but can be proved analogously using φ_4 and φ_5 in Condition 2 of Definition 1.

This finishes the proof of the Claim.

To finish the proof, it remains to show that $\mathfrak{I}$ is a model of $\mathcal{K}$.

We first show that $\mathfrak{I} \models \mathcal{T}$. Fix a $d \in \Delta^{\mathfrak{I}}$ and $\pi(d) = (t, i) \in S_\omega$. By definition of type, $C_{\mathcal{T}} \in t$ and, by construction, $X_{C_{\mathcal{T}}} \in \tau_{t,i}(w) = \theta(d, w)$ for all $w \in W$. Then, by the previous claim, $d \in C_{\mathcal{T}}^{\mathfrak{I},w}$ for all $w \in W$. Therefore, $\mathfrak{I}$ is a model of $\mathcal{T}$.

We finally show that $\mathfrak{I}$ is a model of the ABox $\mathcal{A}$. For all $\mathbf{P}\bigcirc^i A(a), \mathbf{P}\bigcirc^i \neg B(a) \in \mathcal{A}$, $\mathfrak{I} \models \mathbf{P}\bigcirc^i A(a)$ ($\mathfrak{I} \models \mathbf{P}\bigcirc^i \neg B(a)$) follows from the claim above and the initial step of the construction.

$\mathfrak{I} \models \mathbf{P}\bigcirc^i r(a, b) \in \mathcal{A}$ and $\mathfrak{I} \models \mathbf{P}\bigcirc^i \neg r(a, b) \in \mathcal{A}$ follows from the fact that φ_1 and φ_2 in Condition $2(c)$ of Definition 1 are satisfied, and from the last point in the initial step of the construction.

"$\Leftarrow$:' Let $\mathfrak{I} = (\Delta^{\mathfrak{I}}, (\mathcal{I}_w)_{w \in W})$ be a model of $\mathcal{K}$. First, define for every $a \in \mathsf{ind}(\mathcal{A})$ the $2^{\mathsf{acl}(\mathcal{K})}$-tree (T, τ_a) as follows, for all $w \in W$:

$$\tau_a(w) = \{C \mid C \in \mathsf{ccl}(\mathcal{K}) \wedge a \in C^{\mathfrak{I},w}\} \cup \{\mathbf{P}\bigcirc^i r(a, b) \in \mathsf{acl}(\mathcal{K}) \mid \mathfrak{I} \models \mathbf{P}\bigcirc^i r(a, b)\}$$

For all $d \in \Delta^{\mathfrak{I}} \setminus \mathsf{ind}(\mathcal{A})$, we define the $2^{\mathsf{ccl}(\mathcal{K})}$-tree (T, τ_d) as follows, for all $w \in W$:

$$\tau_d(w) = \{C \mid C \in \mathsf{ccl}(\mathcal{K}) \wedge d \in C^{\mathfrak{I},w}\}$$

We now define the following set of types

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

It is routine to verify that S satisfies Conditions 1 and 2 of Definition 1. Since $\mathfrak{I}$ is a model of $\mathcal{K}$, for each $a \in \mathsf{ind}(\mathcal{A})$, there exists $(t_a, 0) \in S$ such that for all $\alpha \in \mathcal{A}$ with a occurring in α we have that $\alpha \in t_a$.

□

Proofs for Section 4

Definition 5 (Semantics of Temporal Formulas)**.** *Let $\mathfrak{I}$ be a temporal interpretation. For a time point w in $\mathfrak{I}$, the* truth relation $\models$ *for temporal CTL*-Lite$_{bool}^{\mathcal{N}}$-state Formulas is defined as follows:*

$$\begin{aligned}
\mathfrak{I}, w \models C \sqsubseteq D \quad & \textit{iff} \quad C^{\mathfrak{I},w} \subseteq D^{\mathfrak{I},w},\\
\mathfrak{I}, w \models C(a) \quad & \textit{iff} \quad a \in C^{\mathfrak{I},w},\\
\mathfrak{I}, w \models r(a, b) \quad & \textit{iff} \quad (a, b) \in r^{\mathfrak{I},w},\\
\mathfrak{I}, w \models \neg\varphi \quad & \textit{iff} \quad \mathfrak{I}, w \not\models \varphi,\\
\mathfrak{I}, w \models \varphi_1 \wedge \varphi_2 \quad & \textit{iff} \quad \mathfrak{I}, w \models \varphi_1 \textit{ and } \mathfrak{I}, w \models \varphi_2,\\
\mathfrak{I}, w \models \mathbf{E}\psi \quad & \textit{iff} \quad \mathfrak{I}, \pi \models \psi \textit{ for some } \pi \in \mathsf{Paths}(w).
\end{aligned}$$

For a path π in $\mathfrak{I}$, the truth relation $\models$ for path formulas is defined as follows:

$$\begin{aligned}
\mathfrak{I}, \pi \models \varphi \quad & \textit{iff} \quad \mathfrak{I}, \pi[0] \models \varphi,\\
\mathfrak{I}, \pi \models \neg\psi \quad & \textit{iff} \quad \mathfrak{I}, \pi \not\models \psi,\\
\mathfrak{I}, \pi \models \psi_1 \wedge \psi_2 \quad & \textit{iff} \quad \mathfrak{I}, \pi \models \psi_1 \textit{ and } \mathfrak{I}, \pi \models \psi_2,\\
\mathfrak{I}, \pi \models \bigcirc\psi \quad & \textit{iff} \quad \mathfrak{I}, \pi[1..] \models \psi,\\
\mathfrak{I}, \pi \models \Box\psi \quad & \textit{iff} \quad \forall j \geq 0.\pi[j..] \models \psi,\\
\mathfrak{I}, \pi \models \psi_1 \mathcal{U} \psi_2 \quad & \textit{iff} \quad \exists j \geq 0.(\mathfrak{I}, \pi[j..] \models \psi_2 \wedge\\
& \qquad \forall 0 \leq k < j.(\mathfrak{I}, \pi[k..] \models \psi_1)).
\end{aligned}$$

We say that a temporal interpretation $\mathfrak{I}$ is a *model* of a temporal CTL$^*_{\mathcal{ALC}}$-TBox φ if $\mathfrak{I}, \varepsilon \models \varphi$.

Lemma 1 *φ is satis. iff there is a proper quasimodel for φ*

Proof Sketch. "$\Rightarrow$" Let $\mathfrak{I} = (\Delta^{\mathfrak{I}}, (\mathcal{I}_w)_{w \in W})$ be al model of φ. We define a $\mathsf{qs}(w)$-labeled tree structure $\mathfrak{Q} = (T, \tau)$ such that for all $w \in T$, $\tau(w)$ is defined as follows:

$$\begin{aligned}
\pi(d, w) &= \{C \in \mathsf{cl}(\varphi) \mid d \in C^{\mathfrak{I},w}\};\\
S_1(w) &= \{\pi(d, w) \mid d \in \Delta\};\\
S_2(w) &= \{(\pi(a, w), a) \mid a \in \mathsf{ind}(\varphi)\};\\
S_3(w) &= \{\psi \in \mathsf{sub}(\varphi) \mid \mathfrak{I}, w \models \psi\}.
\end{aligned}$$

We obtain the 2^{fcn}- labeled tree $\mathfrak{Q}_3$ by associating each $w \in \mathfrak{Q}$ with the label $\bar{S}_3(w)$. Moreover, it is not hard to see that for all $w \in T$ with $\tau(w) = (S_1, S_2, S_3)$ and all $\pi(d, w) \in S_1$ there is a 2^{cnn}-labeled tree (T, τ') satisfying $2(a)$-(c), and for all $a \in \mathsf{ind}(\varphi)$ and $(\pi(a, w), a) \in S_2$ there is a 2^{cnn}-labeled tree (T, τ') satisfying $3(a)$-(c). Then, $\mathfrak{Q}$ is indeed a proper-quasimodel of φ.

"$\Leftarrow$" Let $\mathfrak{Q} = (T, \tau)$ be a proper-quasimodel of φ. According to Condition 2 , for all $w \in T$ with $\tau(w) = (S_1, S_2, S_3)$ and all $t \in S_1$ there is a there is a 2^{ccn}-labeled tree $(T, \tau_{t,w})$ satisfying $2(a)$-(c). Analogously, due to Condition 3, for all $w \in T$ with $\tau(w) = (S_1, S_2, S_3)$ and $a \in \mathsf{ind}(\varphi)$ with $(t, a) \in S_2$ there is a 2^{ccn}-labeled tree $(T, \tau_{t_a,w})$ satisfying $3(a)$-(c). We define the interpretation $\mathfrak{I} = (\Delta^{\mathfrak{I}}, (\mathcal{I}_w)_{w \in W})$ inductively as follows:

We define infinite sequences $\Delta_0, \Delta_1, \ldots$ and mappings $\theta_i : \Delta_i \times W \to 2^{\mathsf{ccn}}$ and relations R_0^r, R_1^r with $r \in \mathsf{Rol}(\varphi)$.

To start the construction of $\mathfrak{I}$, set

- $\Delta_0 = \{(t, w) \mid t \in S_1(w), w \in W\} \cup \mathsf{ind}(\varphi)$;
- For all $(t, v) \in \Delta_0$, set $\theta_0((t, v), w) ::= \tau_{t,v}(w)$;

  For all $a \in \mathsf{ind}(\varphi)$, set $\theta_0(a, w) ::= \tau_{t,\varepsilon}(w)$ such that $(t, a) \in S_2(w)$;[3]
- For $r \in \mathsf{Rol}_{\mathsf{loc}}(\varphi)$, R_0^r is defined as follows

  $$\{(a, b, w) \in (\Delta_0)^2 \times W \mid r(a, b) \in \bar{S}_3(w) \wedge a, b \in \mathsf{ind}(\varphi)\}$$
- For $r \in \mathsf{Rol}_{\mathsf{rig}}(\varphi)$, R_0^r is defined as follows. For all $w' \in W$

  $$\{(a, b, w') \in (\Delta_0)^2 \times W \mid \exists w \in W.\, r(a, b) \in \bar{S}_3(w) \wedge a, b \in \mathsf{ind}(\varphi)\}$$

In the inductive step below, we fix again 'defective' points missing successors. For this purpose, we will use the notions of required and actual rank as in the proof of Theorem 1. The *required r-rank for a $d \in \Delta_i, i \geq 1$, at world w and step i* is defined as follows:

$$\sharp_{d,i}^{r,w} = \mathtt{max}\{n \in \sharp_{\mathcal{T}}^r \mid X_n^r \in \theta_i(d, w)\}^4$$

The *actual r-rank for a $d \in \Delta_i, i \geq 1$, at world w and step i* is defined as follows:

$$\xi_{d,i}^{r,w} = \mathtt{max}\{\{0\} \cup \{n \in \sharp_{\mathcal{T}}^r \mid \exists \text{ distinct } e_1, \ldots e_n \in \Delta_i \wedge (d, e_j) \in R_i^r(w) \text{ or } (e_j, d) \in R_i^{r^-}(w)\}\}$$

For the inductive step, set $\Delta_i = \Delta_{i-1}$, $\theta_i = \theta_{i-1}$, $R_i^r = R_{i-1}^r$ and apply the following, which is meant to fix 'defective' points missing successors:

- If $\sharp_{d,i}^{r,w} - \xi_{d,i}^{r,w} = n > 0$ for some $d \in \Delta_{i-1}$ and $w \in W$ and $r \in \mathsf{Rol}(\mathcal{K})$, then proceed as follows:
  1. Add $e_1, \ldots, e_n$ to Δ_i;
  2. Add (d, e_i, w) to R_i^r if r is local; otherwise, for all $w' \in W$, add (d, e_i, w') to R_i^r;
  3. For all $w' \in W$, set $\theta_i(e_i, w') = \tau_{(t,w)}(w')$ such that $\exists r^- \in t$.

Finally, put $\Delta^{\mathfrak{I}} = \bigcup_{i \geq 0} \Delta_i$, $\theta = \bigcup_{i \geq 0} \theta_i$ and $R^r = \bigcup_{i \geq 0} R_i^r$. It remains to define the interpretation of concept and role names (recall we make the standard name assumption):

$$\begin{array}{rcl} A^{\mathfrak{I},w} & = & \{d \in \Delta^{\mathfrak{I}} \mid A \in \theta(d, w)\}; \\ s^{\mathfrak{I},w} & = & \{(d, e) \in (\Delta^{\mathfrak{I}})^2 \mid (d, e, w) \in R^s\} \cup \\ & & \{(d, e) \in (\Delta^{\mathfrak{I}})^2 \mid (e, d, w) \in R^{s^-}\} \end{array}$$

By using the properties of a proper-quasimodel, one can prove the following claims, by simultaneous structural induction as in the proof of Theorem 1 above, which imply that $\mathfrak{I}$ is a model of φ.

Claim 1. For all $d \in \Delta^{\mathfrak{I}}$, $w \in W$ and $C \in \mathsf{ccl}(\varphi)$, we have that

$$d \in C^{\mathfrak{I},w} \text{ iff } X_C \in \theta(d, w)$$

Claim 2. For all $\vartheta \in \mathsf{sub}(\varphi)$ and $w \in W$, we have that

$$\mathfrak{Q}_3, w \models \bar{\vartheta} \text{ iff } \mathfrak{I}, w \models \vartheta,$$

for every $\pi \in \mathsf{Paths}(w)$ and path formula ψ, we have

$$\mathfrak{Q}_3, \pi \models \bar{\psi} \text{ iff } \mathfrak{I}, \pi \models \psi.$$

□

Alternating Tree Automata

We need some preliminaries. 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. For $k \in \mathbb{N}$, we define $[k] = \{-1, 0, \ldots, k\}$. 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 $c \in \mathbb{N}$.

An *alternating 2-way Büchi tree automaton (2ABTA)* over Σ-labeled k-ary trees is a tuple $\mathbf{A} = (Q, \Sigma, q_0, \delta, F)$ where Q is a finite set of *states*, $q_0 \in Q$ is an *initial state*, δ is the *transition function* $\delta : Q \times \Sigma \times \{\mathsf{t}, \mathsf{f}\} \to \mathcal{B}^+([k] \times Q)$ and $F \subseteq Q$ is the set of *recurring states*. 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 $\mathbf{A}$ on τ is a $T \times Q$-labeled tree (T_r, r) such that $r(\varepsilon) = (\varepsilon, q_0)$ 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 $\tau_r(x \cdot i) = (\mathsf{mov}(w, m_i), q_i)$. Using the 'root flag' as an additional third component in the transition function is non-standard, but it does not cause any problems. We use it to define a more compact 2ABTA below.

Theorem 4 *Satisfiability is in 3*EXPTIME *for CTL*-Lite$_{bool}^{\mathcal{N}}$- and in 2*EXPTIME *for CTL-Lite$_{bool}^{\mathcal{N}}$-formulas*

[3] Recall that by Definition 2 such (t, a) is unique

[4] By convention $n = 0$ if no such proposition is in $\theta(d, w)$

Proof Sketch. We next formally define a 2ABTA $\mathfrak{A}' = (Q, \Sigma, \delta, q_0, F)$ simulating a run of $\mathfrak{A}_\vartheta$ for every $w \in T$ and $t \in S_1$ such that $\tau(w) = (S_1, S_2, S_3)$.

Set $\Sigma = \mathsf{qs}(\varphi)$, $F = F_1$, $Q_1^* = Q_1 \cup \{*\}$ and

$$Q = \{q_0\} \cup (Q_1 \times Q_1^*) \cup (Q_1 \times 2^{\mathsf{cnn}} \times Q_1^*)$$

For all $\sigma = (S_1, S_2, S_3) \in \Sigma$, the transition relation δ is defined as follows:

$$\delta(q_0, \sigma, \cdot) = \bigwedge_{i=1}^{k}(i, q_0) \wedge \bigwedge_{t \in S_1} \bigvee_{q \in Q_1}(0, (q, t, *))$$

$$\delta((q, q'), \sigma, \cdot) = \bigvee_{t \in S_1}(0, (q, t, q'))$$

$$\delta((q, t, q'), \sigma, \mathsf{t}) = \Theta$$

$$\delta((q, t, q'), \sigma, \mathsf{f}) = \bigvee_{q'' \in Q_1}(-1, (q'', q'))) \wedge \Theta$$

where $\Theta = \bigvee_{(q_1, \ldots, q_k) \in \delta_1(q,t) | q' \in \{q_1, \ldots, q_k\}} \bigwedge_{i=1}^{k}(i, (q_i, *))$, '$\cdot$' means that the transition is triggered with both t and f, and '$*$' is used as a placeholder for all states $q \in Q_1$ with $q \in \{q_1, \ldots, q_k\}$.

In the definition of δ, states of the form (q, q') and (q, t, q') use the q' component to help 'putting together' a run of $\mathfrak{A}_\vartheta$ that starts from the root, while initiating a simulation from an arbitrary node. The intuitive reading is that a run of $\mathfrak{A}_\vartheta$ is currently being simulated in state q, and q' has been assigned to some successor node. Additionally, (q, t, q') takes care of choosing a $t \in S_1$ at the current node, as required by Condition 2 in Definition 3. The definition of δ also makes clear why the simulated $\mathfrak{A}_\vartheta$ needs to be a NBTA (rather than an ABTA): one state needs to be assigned to each successor (cf. Condition 2(b)).

As the last step in the construction of $\mathfrak{A}$, since 2ABTAs are trivially closed under intersection, it remains to describe how to construct 2ABTAs checking Conditions 1 and 3: (i) a 2ABTA to check Condition 1 is obtained by manipulating the NBTA $\mathfrak{A}_\psi$ accepting the models (2^{fcn}-labeled trees) of $\psi = \bar{\varphi} \wedge \mathbf{A}\Box(\varphi_1 \wedge \varphi_2)$, so that it has input alphabet $\mathsf{qs}(\varphi)$ and each symbol (S_1, S_2, S_3) is handled as $\bar{S}_3$, and (ii) a 2ABTA to check Condition 3 is obtained as a simpler variant of $\mathfrak{A}'$. Note that we assume wlog. (cf. appendix) that the automata above and $\mathfrak{A}_\vartheta$ in $\mathfrak{A}'$ run on trees of outdegree k (cf. Lemma 2) - this assumption does not affect Q_1 above, and thus k.

Overall Complexity. The upper bounds follow from the following facts: (i) Non-emptiness of 2ABTAs can be checked in EXPTIME in the number of states [Vardi, 1998]. (ii) The number of states of $\mathfrak{A}_\vartheta$, which is the most dominating in the definition of Q in $\mathfrak{A}'$ (and thus also in $\mathfrak{A}$), is in $2^{2^{\mathsf{poly}(|\varphi|)}}$ if ϑ is a CTL* formula and in $2^{\mathsf{poly}(|\varphi|)}$ if ϑ is a CTL formula.

□

Alternating Turing Machines

An *Alternating Turing Machine (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[5]. 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 [Chandra *et al.*, 1981], the problem of deciding whether $w \in L(\mathcal{M})$ is 2EXPTIME-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 $|\, uu' \,| \leq 2^k$.

Theorem 5 *Satisfiability of CTL-Lite$_{horn}$-formulas without rigid roles is 2EXPTIME-hard*

Proof. The upper bounds follow from Theorem 4. We next show that satisfiability of CTL-*Lite*$_{horn}$-formulas is

[5] As this case is simpler than the general one, we define acceptance for ATMs with finite computation paths only, and refer to [Chandra *et al.*, 1981] for the full definition.

2ExpTime-hard, by reducing from the word problem for exponentially space bounded alternating Turing machines (ATM).

Let $\mathcal{M} = (Q, \Sigma, \Gamma, q_0, \delta)$ be such an ATM with a 2ExpTime-hard word problem, and $\omega = a_0 \dots a_{n-1} \in \Sigma$ the input of length n. We construct in polynomial time a CTL-*Lite*$_{horn}$ formula $\varphi_{\mathcal{M},\omega}$ such that $\varphi_{\mathcal{M},\omega}$ is satisfiable iff $\mathcal{M}$ accepts ω. In what follows, we assume that triples α in $\delta(q,a)$ are linearly ordered such that n_α denotes the number of α in the ordering (starting from 0). Let $u ::= \mathtt{max}\{\delta(q,a) \mid q \in Q \wedge a \in \Sigma\}$. We assume wlog. $q_0 \in Q_\exists$.

We outline the main ideas of the reduction: (i) the computation tree of an ATM is represented by the temporal evolution of a single individual name a, such that each time point corresponds to a tape cell and a configuration is then represented by exponentially consecutive time points. (ii) To synchronise i-cells in consecutive configurations, the 'content of a' at that position i is stored in a fresh *local* r_a-successor and then recovered back to a using temporal TBoxes of the form $\top \sqsubseteq A \vee \top \sqsubseteq \bar{A}$, with A and $\bar{A}$ disjoint, ensuring that *all* domain elements in a given time point share the truth value of A. (iii) Information is transported between neighbouring configurations using binary counters.

We proceed now with the reduction. Throughout the reduction, we use various 2^n-counters over the temporal dimensions. We next exemplify how to implement a counter X (1)-(5). To this aim, we use concepts $X_0, \overline{X_0} \dots, X_{n-1}, \overline{X_{n-1}}$ simulating the bits of a number in binary.

For every, $0 \leq j < i < n$

$$\overline{X_i} \sqcap \overline{X_j} \sqsubseteq \mathbf{A}\bigcirc\overline{X_i}, \tag{1}$$

$$X_i \sqcap \overline{X_j} \sqsubseteq \mathbf{A}\bigcirc X_i, \tag{2}$$

For every $0 \leq j < n$,

$$\overline{X_j} \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 \sqsubseteq \mathbf{A}\bigcirc X_j, \tag{3}$$

$$X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 \sqsubseteq \mathbf{A}\bigcirc\overline{X_j}. \tag{4}$$

For every $0 \leq i < n$

$$X_i \sqcap \overline{X_i} \sqsubseteq \bot \tag{5}$$

We use abbreviations Zero and End to respectively denote

$$\prod_{j=0}^{n-1} \overline{X_j} \qquad \prod_{j=0}^{n-1} X_j$$

As the next step, we enforce basic structural requirements of ATMs (6)- (8). To this aim, we use the following signature

- Concept names A_a for each $a \in \Gamma$;
- Concept names Q_q, for each $q \in Q$, to denote the current state and the position of the head;
- Concept $\overline{H}$, marking in a configuration all the cells to the right of the head;
- Concept names $M_\alpha, \mathfrak{N}_\alpha$ for every $\alpha \in \Xi$, where $\Xi = \{(q,a,m) \mid (q',b,q,a,m) \in \delta \text{ for any } b \in \Gamma \wedge q' \in Q\}$;
- Concept names $N^{q,a}_{n_\alpha}$, $\mathfrak{S}^{q,a}_{n_\alpha}$, for every $a \in \Gamma$, $q \in Q \setminus \{q_a, q_r\}$ and $\alpha \in \delta(q,a)$;
- Local role names r_a for each $a \in \Gamma$.

To define the desired requirements, we use a counter *Tape* over a configuration of $\mathcal{M}$; in particular, we use abbreviations $\mathsf{ZeroTape}$ and $\mathsf{EndTape}$ as above. First, we ensure that in each configuration the head position is labelled with at most one state variable, and that each cell is labelled with exactly one alphabet letter (6). Furthermore, we require that in each configuration at most one tape cell is labelled with a state Q_q (8). To this aim, we mark all cells to right (until we reach the end of the tape) of the head position with $\overline{H}$ (7).

$$\bigwedge_{q \neq q' \in Q} (Q_q \sqcap Q_{q'} \sqsubseteq \bot) \wedge \bigwedge_{a \neq a' \in \Gamma} (A_a \sqcap A_{a'} \sqsubseteq \bot) \tag{6}$$

$$\bigwedge_{q \in Q} (Q_q \sqsubseteq \mathbf{A}\bigcirc\overline{H}) \wedge \bigwedge_{j=0}^{n-1} (\overline{H} \sqcap \overline{Tape_j} \sqsubseteq \mathbf{A}\bigcirc\overline{H}) \tag{7}$$

$$\bigwedge_{q \in Q} (\overline{H} \sqcap Q_q \sqsubseteq \bot) \tag{8}$$

We will further use a counters *Head* and *Head'*, which stop when reaching the end of the tape.

$$\bigwedge_{j=0}^{n-1} (\overline{Tape_j} \sqcap \overline{X_i} \sqcap \overline{X_j} \sqsubseteq \mathbf{A}\bigcirc\overline{X_i}), \tag{9}$$

$$\bigwedge_{j=0}^{n-1} (\overline{Tape_j} \sqcap X_i \sqcap \overline{X_j} \sqsubseteq \mathbf{A}\bigcirc X_i), \tag{10}$$

For every $0 \leq j < n$,

$$\bigwedge_{j=0}^{n-1} (\overline{Tape_j} \sqcap \overline{X_j} \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 \sqsubseteq \mathbf{A}\bigcirc X_j), \tag{11}$$

$$\bigwedge_{j=0}^{n-1} (\overline{Tape_j} \sqcap X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 \sqsubseteq \mathbf{A}\bigcirc\overline{X_j}). \tag{12}$$

We next use the counter *Head*, concepts M_α and concepts $N^{q,a}_{n_\alpha}$ storing the information generated by the transition function (13), which will be used to establish the successor configuration. In particular, we carry this information to the end of the tape (14).

$$\bigwedge_{a \in \Gamma, q \in Q \setminus \{q_a, q_r\}} (A_a \sqcap Q_q \sqsubseteq \prod_{\alpha \in \delta(q,a)} \mathbf{E}\bigcirc(M_\alpha \sqcap N^{q,a}_{n_\alpha}) \sqcap \mathsf{ZeroHead}) \tag{13}$$

$$\bigwedge_{j=0}^{n-1} (M_\alpha \sqcap N^{q,a}_{n_\alpha} \sqcap \overline{Tape_j} \sqsubseteq \mathbf{A}\bigcirc(M_\alpha \sqcap N^{q,a}_{n_\alpha})) \tag{14}$$

We use copies $\mathfrak{N}_{q,a,m}$ of $M_{q,a,m}$, $\mathfrak{S}^{q,a}_{n_\alpha}$ of $N^{q,a}_{n_\alpha}$ and *Head'* of the counter *Head* to avoid clashes while synchronising neighbouring configurations (15)-(16). In (16), we use $\mathbf{A}\bigcirc$ $(C_{Head'} = C_{Head} + 1 \mod 2^n)$ to denote that the value of the *Head'*-counter in all successor worlds is equal to the value of the $Head$-counter (in the current world) plus 1 modulo 2^n; this can be implemented by recasting the incrementation axioms given above. We transport the copies until the end of the tape (17).

$$\bigwedge_{\alpha \in \Xi} (M_\alpha \sqcap \mathsf{EndTape} \sqcap N^{q,a}_{n_\alpha} \sqsubseteq \mathbf{E}\bigcirc(\mathfrak{N}_\alpha \sqcap \mathfrak{S}^{q,a}_{n_\alpha})) \tag{15}$$

$$\mathsf{EndTape} \sqsubseteq \mathbf{A}\bigcirc\,(C_{Head'} = C_{Head} + 1 \mod 2^n) \tag{16}$$

$$\bigwedge_{\alpha \in \Xi} \bigwedge_{j=0}^{n-1} (\mathfrak{N}_\alpha \sqcap \mathfrak{S}^{q,a}_{n_\alpha} \sqcap \overline{Tape_j} \sqsubseteq \mathbf{A}\bigcirc(\mathfrak{N}_\alpha \sqcap \mathfrak{S}^{q,a}_{n_\alpha})) \tag{17}$$

$$\bigwedge_{\alpha \neq \alpha' \in \Xi} ((\mathfrak{N}_\alpha \sqcap \mathfrak{N}_{\alpha'} \sqsubseteq \bot) \wedge (M_\alpha \sqcap M_{\alpha'} \sqsubseteq \bot)) \tag{18}$$

We next describe the changes imposed by the transition relation for elements under the head. We particularly ensure that the new tape symbol is written, the state variable is set in the correct position and that the head is not pushed beyond the end of the tape.

$$\bigwedge_{(q,a,m) \in \Xi} \mathfrak{N}_{q,a,m} \sqcap \mathsf{ZeroHead}' \sqsubseteq A_a \tag{19}$$

$$\bigwedge_{(q,a,n) \in \Xi} \mathfrak{N}_{q,a,n} \sqcap \mathsf{ZeroHead}' \sqsubseteq Q_q \tag{20}$$

$$\bigwedge_{(q,a,r) \in \Xi} \mathfrak{N}_{q,a,r} \sqcap \mathsf{ZeroHead}' \sqsubseteq \mathbf{A}\bigcirc Q_q \tag{21}$$

$$\bigwedge_{(q,a,l) \in \Xi} \mathfrak{N}_{q,a,r} \sqcap \mathsf{EndHead}' \sqsubseteq Q_q \tag{22}$$

$$\bigwedge_{(q,a,l) \in \Xi} \mathsf{ZeroHead}' \sqcap \mathsf{ZeroTape} \sqsubseteq \overline{\mathfrak{N}_{q,a,m}} \tag{23}$$

We next ensure that cells that are not under the head do not change their contents during the transition. We first store the information of their contents in fresh elements and synchronise them with the previous configuration using a counter $Cell$. To store the content of a cell, we will use concept names C_a and copies B_a, for each $a \in \Gamma$. In (24)-(26) below, we transport such concept names until the end of the previous configuration, and then make copies to avoid clashes.

$$\bigwedge_{a \in \Gamma} \bigwedge_{j=0}^{n-1} (\overline{Tape_j} \sqcap \mathbf{E}\bigcirc B_a \sqsubseteq B_a) \tag{24}$$

$$\bigwedge_{a \in \Gamma} \bigwedge_{j=0}^{n-1} (\overline{Tape_j} \sqcap \mathbf{E}\bigcirc C_a \sqsubseteq C_a) \tag{25}$$

$$\bigwedge_{a \in \Gamma} (\mathsf{EndTape} \sqcap \mathbf{E}\bigcirc C_a \sqsubseteq B_a) \tag{26}$$

$$\bigwedge_{a \neq a' \in \Gamma} (C_a \sqcap C_{a'} \sqsubseteq \bot \wedge B_a \sqcap B_{a'} \sqsubseteq \bot) \tag{27}$$

We proceed to propagate information of cells not meant to change their contents, ensuring they remain the same in neighbouring configurations. To do so, we enforce that in a given time point all elements share the same alphabet symbol (28), and the same value of $Tape$-counter value (29) - used in (24)-(26) over the fresh r_a-elements defined below. For all those cells that are *not* in $\mathsf{ZeroHead}'$, and therefore are not changing, an r_a-representative is generated and labelled with C_a and a $Cell$-counter is initialised for such element (30). Finally, we synchronise the content of each such i-cell with that of the i-cell in the previous configuration (31).

$$\bigwedge_{a \in \Gamma} (\top \sqsubseteq A_a \vee \top \sqsubseteq \overline{A_a}) \tag{28}$$

$$\bigwedge_{j=0}^{n-1} (\top \sqsubseteq Tape_j \vee \top \sqsubseteq \overline{Tape_j}) \tag{29}$$

$$\bigwedge_{a \in \Gamma} (\bigwedge_{j=0}^{n-1} (Head'_j \sqcap A_a \sqsubseteq \exists r_a) \wedge (\exists r_a^- \sqsubseteq C_a \sqcap \mathsf{ZeroCell})) \tag{30}$$

$$\bigwedge_{a \neq a' \in \Gamma} (A_a \sqcap B_{a'} \sqcap \mathsf{ZeroCell} \sqsubseteq \bot) \tag{31}$$

We next identify accepting configurations in a bottom-up manner. Intuitively, we will propagate bottom-up the fact that we have reached an (Acc_*) accepting configuration at the leaves (32)-(33). We will generally use two types of Acc-like markers to differentiate whether we are dealing with a universal or existential state; and will copy the markers until reaching the end of the tape of the previous configuration and then make copies (34)-(35). In the inner nodes, we will then proceed by differentiating whether we are looking (i) at a universal state and are located at a successor configuration of a universal (36) or existential state (37); or (ii) at a existential state and are located at a successor configuration of a universal (38) or existential state (39). For all $q_1, q_2 \in Q_\forall$, $q'_1, q'_2 \in Q_\exists$, $a_1, a_2 \in \Sigma$, $\alpha \in \delta(q_i, a_j)$ and $\alpha' \in \delta(q'_i, a_j)$, we define the following.

$$\mathfrak{S}^{q_1,a_1}_{n_\alpha} \sqcap q_a \sqsubseteq \mathsf{Acc}_{n_\alpha} \tag{32}$$

$$\mathfrak{S}^{q'_1,a_1}_{n_{\alpha'}} \sqcap q_a \sqsubseteq \mathsf{Acc} \tag{33}$$

$$\begin{aligned} \textstyle\bigwedge_{j=0}^{n-1} ((&\overline{Tape_j} \sqcap \mathbf{E}\bigcirc\mathsf{Acc} \sqsubseteq \mathsf{Acc}) \wedge \\ &(\overline{Tape_j} \sqcap \mathbf{E}\bigcirc\mathsf{Acc}_{n_\alpha} \sqsubseteq \mathsf{Acc}_{n_\alpha})) \end{aligned} \tag{34}$$

$$\begin{aligned} (&\mathsf{EndTape} \sqcap \mathbf{E}\bigcirc\mathsf{Acc} \sqsubseteq \mathsf{Acc}') \wedge \\ &(\mathsf{EndTape} \sqcap \mathbf{E}\bigcirc\mathsf{Acc}_{n_\alpha} \sqsubseteq \mathsf{Acc}'_{n_\alpha}) \end{aligned} \tag{35}$$

$$A_{a_1} \sqcap Q_{q_1} \sqcap \bigsqcap_{k < \sharp\delta(q_1,a_1)} \mathsf{Acc}'_k \sqcap \mathfrak{S}^{q_2,a_2}_{n_\alpha} \sqsubseteq \mathsf{Acc}_{n_\alpha} \tag{36}$$

$$A_{a_1} \sqcap Q_{q_1} \sqcap \bigsqcap_{k < \sharp\delta(q_1,a_1)} \mathsf{Acc}'_k \sqcap \mathfrak{S}^{q'_1,a_2}_{n_{\alpha'}} \sqsubseteq \mathsf{Acc} \tag{37}$$

$$Q_{q'_1} \sqcap \mathsf{Acc}' \sqcap \mathfrak{S}^{q_1,a_2}_{n_\alpha} \sqsubseteq \mathsf{Acc}_{n_\alpha} \tag{38}$$

$$Q_{q'_1} \sqcap \mathsf{Acc}' \sqcap \mathfrak{S}^{q'_2,a_2}_{n_{\alpha'}} \sqsubseteq \mathsf{Acc} \tag{39}$$

We next define the initial configuration. For $1 \leq i < n$

$$\begin{aligned} A_0 \sqsubseteq A_{a_0} \sqcap Q_{q_0} \sqcap \mathsf{ZeroTape} \sqcap \mathbf{E}\bigcirc A_1 \\ A_i \sqsubseteq A_{a_i} \sqcap \mathbf{E}\bigcirc A_{i+1} \\ A_n \sqsubseteq A_{\llcorner\!\lrcorner} \\ \bigwedge_{j=0}^{n-1} (\overline{Tape_j} \sqcap A_{\llcorner\!\lrcorner} \sqsubseteq \mathbf{E}\bigcirc A_{\llcorner\!\lrcorner}) \end{aligned} \tag{40}$$

We define ψ as the conjunction of the TBoxes above and $\varphi_{\mathcal{M},\omega}$ as $\mathbf{A}\Box\psi \wedge (A_0 \sqcap \mathsf{Acc}')(a)$. Finally, following the intuitive meaning of each conjunct above, it is not hard to see that $\varphi_{\mathcal{M},\omega}$ is satisfiable iff $\mathcal{M}$ accepts ω.

This finishes the proof. $\square$